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.

<table>
<thead>
<tr>
<th>Site code</th>
<th>Island size (ha)</th>
<th>Number of recorders</th>
<th>Sampling effort</th>
<th>Observed richness</th>
<th>Rarefied richness</th>
<th>Sample coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.45</td>
<td>1</td>
<td>62</td>
<td>5</td>
<td>5.000</td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>0.45</td>
<td>1</td>
<td>62</td>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>3</td>
<td>0.63</td>
<td>1</td>
<td>62</td>
<td>2</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>4</td>
<td>0.63</td>
<td>1</td>
<td>59</td>
<td>4</td>
<td>4.000</td>
<td>1.000</td>
</tr>
<tr>
<td>5</td>
<td>0.63</td>
<td>1</td>
<td>62</td>
<td>0</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>6</td>
<td>0.72</td>
<td>1</td>
<td>62</td>
<td>5</td>
<td>5.000</td>
<td>1.000</td>
</tr>
<tr>
<td>7</td>
<td>0.91</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>8</td>
<td>0.91</td>
<td>1</td>
<td>62</td>
<td>9</td>
<td>9.000</td>
<td>1.000</td>
</tr>
<tr>
<td>9</td>
<td>1.27</td>
<td>1</td>
<td>62</td>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>10</td>
<td>1.36</td>
<td>1</td>
<td>62</td>
<td>2</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>11</td>
<td>1.45</td>
<td>1</td>
<td>62</td>
<td>2</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>12</td>
<td>1.54</td>
<td>1</td>
<td>62</td>
<td>7</td>
<td>7.000</td>
<td>1.000</td>
</tr>
<tr>
<td>13</td>
<td>1.81</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>14</td>
<td>1.99</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>15</td>
<td>2.08</td>
<td>1</td>
<td>62</td>
<td>2</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>16</td>
<td>2.08</td>
<td>1</td>
<td>62</td>
<td>4</td>
<td>4.000</td>
<td>1.000</td>
</tr>
<tr>
<td>17</td>
<td>2.08</td>
<td>1</td>
<td>62</td>
<td>9</td>
<td>9.000</td>
<td>1.000</td>
</tr>
<tr>
<td>18</td>
<td>2.26</td>
<td>1</td>
<td>62</td>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>19</td>
<td>2.63</td>
<td>1</td>
<td>62</td>
<td>7</td>
<td>7.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Site code</td>
<td>Island size (ha)</td>
<td>Number of recorders</td>
<td>Sampling effort</td>
<td>Observed richness</td>
<td>Rarefied richness</td>
<td>Sample coverage</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------</td>
<td>---------------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>20</td>
<td>3.08</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>21</td>
<td>3.53</td>
<td>1</td>
<td>62</td>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>22</td>
<td>3.53</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>0.508</td>
</tr>
<tr>
<td>23</td>
<td>3.62</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>24</td>
<td>4.17</td>
<td>1</td>
<td>62</td>
<td>4</td>
<td>4.000</td>
<td>1.000</td>
</tr>
<tr>
<td>25</td>
<td>4.98</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>26</td>
<td>5.43</td>
<td>1</td>
<td>62</td>
<td>2</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>27</td>
<td>5.61</td>
<td>1</td>
<td>62</td>
<td>4</td>
<td>4.000</td>
<td>1.000</td>
</tr>
<tr>
<td>28</td>
<td>6.79</td>
<td>1</td>
<td>62</td>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>29</td>
<td>6.88</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>30</td>
<td>7.43</td>
<td>1</td>
<td>62</td>
<td>5</td>
<td>5.000</td>
<td>1.000</td>
</tr>
<tr>
<td>31</td>
<td>8.15</td>
<td>1</td>
<td>62</td>
<td>7</td>
<td>7.000</td>
<td>0.980</td>
</tr>
<tr>
<td>32</td>
<td>8.15</td>
<td>1</td>
<td>62</td>
<td>7</td>
<td>7.000</td>
<td>1.000</td>
</tr>
<tr>
<td>33</td>
<td>8.42</td>
<td>1</td>
<td>62</td>
<td>5</td>
<td>5.000</td>
<td>1.000</td>
</tr>
<tr>
<td>34</td>
<td>8.78</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>35</td>
<td>9.42</td>
<td>1</td>
<td>62</td>
<td>6</td>
<td>6.000</td>
<td>1.000</td>
</tr>
<tr>
<td>36</td>
<td>11.96</td>
<td>1</td>
<td>62</td>
<td>5</td>
<td>5.000</td>
<td>1.000</td>
</tr>
<tr>
<td>37</td>
<td>13.04</td>
<td>1</td>
<td>62</td>
<td>7</td>
<td>7.000</td>
<td>0.988</td>
</tr>
<tr>
<td>38</td>
<td>13.22</td>
<td>1</td>
<td>60</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>39</td>
<td>13.31</td>
<td>1</td>
<td>62</td>
<td>9</td>
<td>9.000</td>
<td>1.000</td>
</tr>
<tr>
<td>40</td>
<td>15.67</td>
<td>1</td>
<td>62</td>
<td>2</td>
<td>2.000</td>
<td>1.000</td>
</tr>
<tr>
<td>41</td>
<td>16.94</td>
<td>1</td>
<td>62</td>
<td>10</td>
<td>10.000</td>
<td>1.000</td>
</tr>
<tr>
<td>42</td>
<td>17.57</td>
<td>1</td>
<td>62</td>
<td>8</td>
<td>8.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Site code</td>
<td>Island size (ha)</td>
<td>Number of recorders</td>
<td>Sampling effort</td>
<td>Observed richness</td>
<td>Rarefied richness</td>
<td>Sample coverage</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------</td>
<td>---------------------</td>
<td>-----------------</td>
<td>------------------</td>
<td>------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>43</td>
<td>17.66</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>44</td>
<td>21.37</td>
<td>1</td>
<td>62</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>45</td>
<td>22.01</td>
<td>1</td>
<td>62</td>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>46</td>
<td>29.62</td>
<td>2</td>
<td>124</td>
<td>8</td>
<td>7.253</td>
<td>0.972</td>
</tr>
<tr>
<td>47</td>
<td>32.78</td>
<td>2</td>
<td>124</td>
<td>4</td>
<td>3.251</td>
<td>0.928</td>
</tr>
<tr>
<td>48</td>
<td>32.87</td>
<td>2</td>
<td>124</td>
<td>1</td>
<td>0.752</td>
<td>0.504</td>
</tr>
<tr>
<td>49</td>
<td>35.60</td>
<td>2</td>
<td>93</td>
<td>5</td>
<td>4.996</td>
<td>0.999</td>
</tr>
<tr>
<td>50</td>
<td>35.87</td>
<td>2</td>
<td>124</td>
<td>5</td>
<td>4.924</td>
<td>0.991</td>
</tr>
<tr>
<td>51</td>
<td>38.94</td>
<td>2</td>
<td>124</td>
<td>3</td>
<td>3.000</td>
<td>1.000</td>
</tr>
<tr>
<td>52</td>
<td>39.12</td>
<td>2</td>
<td>124</td>
<td>4</td>
<td>3.927</td>
<td>0.992</td>
</tr>
<tr>
<td>53</td>
<td>39.67</td>
<td>2</td>
<td>124</td>
<td>7</td>
<td>5.756</td>
<td>0.948</td>
</tr>
<tr>
<td>54</td>
<td>39.94</td>
<td>2</td>
<td>124</td>
<td>9</td>
<td>8.250</td>
<td>0.988</td>
</tr>
<tr>
<td>55</td>
<td>50.08</td>
<td>2</td>
<td>124</td>
<td>7</td>
<td>6.249</td>
<td>0.986</td>
</tr>
<tr>
<td>56</td>
<td>52.71</td>
<td>2</td>
<td>124</td>
<td>4</td>
<td>3.000</td>
<td>0.982</td>
</tr>
<tr>
<td>57</td>
<td>53.35</td>
<td>2</td>
<td>118</td>
<td>5</td>
<td>3.827</td>
<td>0.966</td>
</tr>
<tr>
<td>58</td>
<td>70.55</td>
<td>2</td>
<td>124</td>
<td>8</td>
<td>7.490</td>
<td>0.984</td>
</tr>
<tr>
<td>59</td>
<td>77.80</td>
<td>2</td>
<td>124</td>
<td>10</td>
<td>9.456</td>
<td>0.993</td>
</tr>
<tr>
<td>60</td>
<td>91.30</td>
<td>3</td>
<td>186</td>
<td>8</td>
<td>7.999</td>
<td>1.000</td>
</tr>
<tr>
<td>61</td>
<td>108.76</td>
<td>3</td>
<td>186</td>
<td>11</td>
<td>10.483</td>
<td>0.987</td>
</tr>
<tr>
<td>62</td>
<td>171.73</td>
<td>3</td>
<td>186</td>
<td>12</td>
<td>10.401</td>
<td>0.986</td>
</tr>
<tr>
<td>63</td>
<td>198.52</td>
<td>3</td>
<td>132</td>
<td>8</td>
<td>7.011</td>
<td>0.936</td>
</tr>
<tr>
<td>64</td>
<td>217.63</td>
<td>3</td>
<td>186</td>
<td>13</td>
<td>10.618</td>
<td>0.977</td>
</tr>
<tr>
<td>65</td>
<td>230.70</td>
<td>3</td>
<td>186</td>
<td>9</td>
<td>8.421</td>
<td>0.993</td>
</tr>
<tr>
<td>Site code</td>
<td>Island size (ha)</td>
<td>Number of recorders</td>
<td>Sampling effort</td>
<td>Observed richness</td>
<td>Rarefied richness</td>
<td>Sample coverage</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------</td>
<td>---------------------</td>
<td>-----------------</td>
<td>------------------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>66</td>
<td>232.49</td>
<td>3</td>
<td>186</td>
<td>11</td>
<td>10.432</td>
<td>0.989</td>
</tr>
<tr>
<td>67</td>
<td>336.02</td>
<td>3</td>
<td>186</td>
<td>9</td>
<td>8.382</td>
<td>0.990</td>
</tr>
<tr>
<td>68</td>
<td>466.60</td>
<td>3</td>
<td>186</td>
<td>12</td>
<td>11.034</td>
<td>0.986</td>
</tr>
<tr>
<td>69</td>
<td>499.91</td>
<td>4</td>
<td>248</td>
<td>13</td>
<td>11.069</td>
<td>0.970</td>
</tr>
<tr>
<td>70</td>
<td>638.66</td>
<td>5</td>
<td>310</td>
<td>18</td>
<td>15.095</td>
<td>0.975</td>
</tr>
<tr>
<td>71</td>
<td>668.03</td>
<td>7</td>
<td>434</td>
<td>18</td>
<td>13.491</td>
<td>0.976</td>
</tr>
<tr>
<td>72</td>
<td>941.71</td>
<td>4</td>
<td>248</td>
<td>16</td>
<td>11.685</td>
<td>0.958</td>
</tr>
<tr>
<td>73</td>
<td>1350.56</td>
<td>4</td>
<td>248</td>
<td>21</td>
<td>16.494</td>
<td>0.969</td>
</tr>
<tr>
<td>74</td>
<td>1698.84</td>
<td>4</td>
<td>248</td>
<td>13</td>
<td>10.998</td>
<td>0.969</td>
</tr>
<tr>
<td>75</td>
<td>CF</td>
<td>10</td>
<td>614</td>
<td>20</td>
<td>13.355</td>
<td>0.937</td>
</tr>
<tr>
<td>76</td>
<td>CF</td>
<td>5</td>
<td>310</td>
<td>14</td>
<td>12.219</td>
<td>0.975</td>
</tr>
<tr>
<td>77</td>
<td>CF</td>
<td>4</td>
<td>248</td>
<td>13</td>
<td>11.419</td>
<td>0.969</td>
</tr>
<tr>
<td>78</td>
<td>CF</td>
<td>4</td>
<td>248</td>
<td>16</td>
<td>12.676</td>
<td>0.962</td>
</tr>
</tbody>
</table>
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).

<table>
<thead>
<tr>
<th>Family</th>
<th>Species</th>
<th>Sites</th>
<th>Reproductive mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Craugastoridae</td>
<td><em>Pristimantis fenestratus</em></td>
<td>10</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td><em>Pristimantis ockendeni</em></td>
<td>18</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td><em>Pristimantis zimmermanae</em></td>
<td>46</td>
<td>23</td>
</tr>
<tr>
<td>Bufonidae</td>
<td><em>Atelopus hoogmoedi</em></td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Rhaebo guttatus</em></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Rhinella marina</em></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Rhinella merianae</em></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Ceratophryidae</td>
<td><em>Ceratophrys cornuta</em></td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Aromobatidae</td>
<td><em>Anomaloglossus stepheni</em></td>
<td>28</td>
<td>21</td>
</tr>
<tr>
<td>Dendrobatidae</td>
<td><em>Ameerega hahneli</em></td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td><em>Ameerega trivittata</em></td>
<td>54</td>
<td>20</td>
</tr>
<tr>
<td>Hylidae</td>
<td><em>Boana boans</em></td>
<td>34</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td><em>Boana calcarata</em></td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Boana cinerascens</em></td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Boana lanciformis</em></td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Boana aff. fasciata</em></td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Dendropsophus brevifrons</em></td>
<td>39</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td><em>Dendropsophus minusculus</em></td>
<td>45</td>
<td>1</td>
</tr>
<tr>
<td>Family</td>
<td>Species</td>
<td>Sites</td>
<td>Reproductive mode</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------</td>
<td>-------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Phyllomedusidae</td>
<td><em>Phyllomedusa tarsius</em></td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td><em>Phyllomedusa vaillantii</em></td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>Leptodactylidae</td>
<td><em>Adenomera andreae</em></td>
<td>35</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td><em>Adenomera hylaedactyla</em></td>
<td>12</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td><em>Leptodactylus knudseni</em></td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td><em>Leptodactylus longirostris</em></td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td><em>Leptodactylus pentadactylus</em></td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td><em>Leptodactylus stenodema</em></td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>Microhylidae</td>
<td><em>Chiasmocleis shudikarensis</em></td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Elachistocleis bicolor</em></td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Synapturanus mirandaribeiroi</em></td>
<td>8</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td><em>Dendropsophus parviceps</em></td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Osteocephalus buckleyi</em></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td><em>Osteocephalus oophagus</em></td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td><em>Osteocephalus taurinus</em></td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Trachycephalus coriaceus</em></td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Trachycephalus resinifictrix</em></td>
<td>2</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td><em>Scinax garbei</em></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><em>Scinax ruber</em></td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
References


<http://research.amnh.org/herpetology/amphibia/index.html>

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.

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

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


Bell, K. E. 2005. Conservation ecology of amphibians and reptiles in Sarapiqui, Costa Rica: forest fragmentation and long-term population change. MSc Thesis. Florida International University, USA.


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.


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


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


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:


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
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.

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

Figure S1

# 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", "Ad Deus",
            "Cafundo", "Cipoal", "Coata", "Formiga", "Furo_de_Santa_Luzia",
            "Fuzaca", "Garrafa", "Gaviao_real", "Jabuti", "Jiquitaia",
            "Joaninha", "Louzivaldo", "Martelo", "Mascote", "Moita",
            "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, 199, 199,
            192, 229, 229, 277, 277, 192, 219, 199, 192, 213, 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))

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")
```
```r
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**

```r
# 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), ]))
```

```r
##
## 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  9.647e-10 ***
## log10(area)  1.6765     0.2163    7.751  4.47e-12 ***
## ---
## Signif. codes:  < 0.001 '***' 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: 4.47e-12
```

```r
# 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(r^{2}*[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
```
Number of species

The number of species recorded depends on the sampling effort – here measured as the number of 1-min recordings. Thus, the number of species increases with sampling effort until the asymptote of the species accumulation curve is reached. Since the sampling effort among sites was different (the larger the site area, the higher the number of recording stations and by extension of 1-min recordings), we calculated the rarefied number of species (to standardise sampling effort) as well as sample coverage to determine how close the observed number of species was to the “true” estimated number of species.

We standardised the sampling effort to the most frequent number of 1-min recordings across surveyed sites (i.e. statistical mode; \( n = 62 \)). We did so because the iNEXT package (Hsieh, Ma, & Chao, 2016) calculates both the interpolated and extrapolated number of species. Accordingly, we used the interpolated, observed and extrapolated number of species for sites with a sampling effort above \( (n = 33) \), equal to \( (n = 43) \) and below \( (n = 2) \) the statistical mode, respectively (hereafter, we refer to them all as the rarefied number of species). Still in the iNEXT package, we calculated sample coverage to assess whether surveyed sites were satisfactorily inventoried with 62 1-min recordings.

Rarefied number of species

Data preparation

# Create recording-by-species matrices
frogs.recording = as.data.frame(tapply(rawdata$occurrence, 
list(rawdata$recording, rawdata$species), sum))

# Species not detected in a recording are real zeros
frogs.recording[is.na(frogs.recording)] = 0

# Remove the column "None"
frogs.recording = frogs.recording[-which(colnames(frogs.recording) == "None")]

# Create a recording-by-species matrix per site (n = 78)
# This format (matrices stored in a list object) is required
# to calculated the rarefied number of species per site using incidence
# (i.e. presence-absence) data
inext.site = split(frogs.recording, rawdata$site[match(rownames(frogs.recording), 
rawdata$recording)])
# Transpose the data frames stored in the list object
# 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

```r
# 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**

```r
# 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)")
```
Correlation between observed and rarefied number of species

# Pearson's product-moment correlation between observed and rarefied number of species

```r
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 < 0.00000000000000022
## 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 species for the site "Toquinho" was assigned to 0 (zero)

```r
obs.raref = data.frame(obs = specnumber(frogs), raref = richness.site$richness)
```

# Draw the graph of the correlation between observed and rarefied number of species

```r
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,
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.obs.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 assign 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")
```
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" into 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  p-value
## 1  2.302099  2.8364991  2.17011698  3.502881  0.49305716  0.00000001903107
## 2  2.857095  2.0359273  1.23228702  2.839568  0.27151241  0.000003684038432128
## 3  3.260687  1.3001936  0.31134217  2.289045  0.09129034  0.01085953769233076
## 4  3.218833  1.6885078  0.09017261  3.286843  0.07797120  0.03891717653921849
## 5  3.442906  0.2327528 -3.67518220  4.140688 -0.04685428  0.90260418883153742

# 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")(b) +
  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 = "#984ea3", se = FALSE,
aes(x = area, y = richness),
data = subset(sar, sar\$area <= 100)) +
geom_smooth(size = 1, method = "lm", colour = "#e41a1c", 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(r^2)[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,
# Draw the graph for continuous forest sites

```r
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())
```
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),]

# Select 25 islands 100 times
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
}

# ISAR for each subset of 25 sites (n = 100)
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)

# 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,
                 172217419, 1083945832, 309442241, 1378459504, 1974162804,
                 1485726889, 1722217419, 1565203937, 1378459504, 1974162804,
                 1485726889, 349682569, 1083945832, 309442241, 1378459504,
                 1974162804, 1485726889, 1722217419, 1565203937, 1378459504,
                 1974162804)

# 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], ])$coefficients[2]
  r2.full.20[i] = summary(lm(richness ~ log10(area),
    # Model fit (r^2)
data = sar.isl[sample.full.20[, i], ]))$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)

# 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,
                 542346566, 99629308, 760242161, 156335171, 1443509149, 602490744, 1365239361,
                 279730440, 182891111, 1948787162, 1446134351, 582764462, 211891494,
                 264602481, 720226263, 115228529, 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, 206347684, 643829539, 945402271, 2124625018, 1690827344,
                 948726833, 456447170, 866335614, 1032176747, 1794769238, 1958945420,
                 2079412266, 1827382989, 645632015, 905251202, 1532502124, 1411979196,
                 1184782471, 725942691, 1212558348, 1930511492, 921147012, 1364134716,
                 1878989525, 721655347, 10415448, 125429553, 1711864330, 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)

# 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, 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, 1084104337, 1527600430, 1115622710, 289274517, 179332826, 1910766374, 626250152, 1642845096, 502806666, 1562003476, 1781175387, 809230528, 490314009, 1402021001, 368756102, 732395523, 84496965, 352270870, 375811602, 1971730604, 146631313, 2060064172, 1198951728, 156389336, 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),
        data = sar.isl[sample.full.10[, i], ])$coefficients[2]
    r2.full.10[i] = summary(lm(richness ~ log10(area),
        data = sar.isl[sample.full.10[, i], ]))$adj.r.squared
    p.full.10[i] = summary(lm(richness ~ log10(area),
        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, 744945280, 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, 127823977, 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, 968444400, 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(sample.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)

# 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]) +
Non-stratified random selection (short range)

`#box.full.r2`

```r
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)")
```

Non-stratified random selection (short range)

`#box.full.slope`

```r
Figure 4c

```
```r
# 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)
```

# 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, 136737426, 957009295, 1989292791, 1853230961, 1911443969, 1261353721, 1027415282, 241034004, 161718608, 1275703625, 2111439781, 1700598712, 1592916179, 43374530, 232151158, 209340666, 1085935855, 285882049, 2091206380, 1682878078, 85903848, 1697635446, 1635861806, 1022456555, 2024697679, 2101397481, 1344500639, 1822426816, 1311649218, 543324895, 2087581645, 526123100, 1680050699, 566775562, 345167024, 1602664518, 64968048, 1639855157, 1241354938, 148552218, 1861526072, 612335669, 1633123308, 1459100985, 1158121107, 580497679, 306604827, 1332928838, 286243345, 2141373777, 2037212311, 1506619446, 521308150, 2052072612, 241882135, 807303067, 393868183, 1032293094, 1926166158, 909554276, 1742475855, 663146752, 1736829215, 423988712, 180014558, 844018892, 1480232952, 1405965368, 313990093, 1332559669, 347747447, 436081704, 1530297566, 447891509, 1992335844, 881663723, 533522389, 1968307086, 719892721, 87469310, 1778822274, 1736369187, 2082516166, 979182080, 1413199227, 489665294, 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),
    data = sar.isl[sample.short.20[, i], ])$coefficients[2]
  r2.short.20[i] = summary(lm(richness ~ log10(area),
    data = sar.isl[sample.short.20[, i], ]))$adj.r.squared
  p.short.20[i] = summary(lm(richness ~ log10(area),
    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)

# 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, 480846291, 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, 419499047, 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)
# Generate and save 100 random numbers
# sample(short.10: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, 1280508566, 1589730837, 1534601435, 146351227,
                  514405340, 1698506167, 154576283, 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, 258105875, 791640343,
                  392187150, 828995879, 761964529, 1665508994, 720215031, 1482990698,
                  774879948, 118779654, 288976470, 925947713, 1968126902)

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

for (i in 1:100) {
  set.seed(sample.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)
# Generate and save 100 random numbers
# sample(1:Machine$integer.max, 100) # This command generated the following values
seed.short.5 = c(181899906, 572588167, 1792177407, 1659406357, 961103864, 1530683178, 1134007730, 937830870, 1066204421, 1804204256, 503063391, 1748500366, 474255682, 121485877, 1348666140, 852326043, 1085653185, 1412958084, 337382050, 1819034604, 347289743, 1832647990, 185514679, 1223109694, 1973347031, 1800567392, 552920802, 830448278, 373898856, 424438329, 235826459, 1805765409, 2046082354, 746323328, 1748767641, 1189739601, 923159733, 1850158210, 1356612017, 1053329651, 1474626586, 1300879879, 2122733924, 2083408007, 536239387, 706011796, 1637424859, 657349614, 11260282, 17681210, 342174539, 1576634915, 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, 602195336, 1792612878, 286096883, 490370245, 2094825162, 2078260721, 1757203739, 1944754230, 1532469947, 1698773672, 753595436, 437582555, 174664625, 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)
# 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$col,
              alpha = 0.25, width = 0.25, height = 0) +
  stat_summary(fun.data = data_summary, colour = "#e41a1c", size = 1) +
  theme_bw(base_size = 20) +
  theme(panel.grid = element_blank(),
        panel.border = element_rect(colour = "black"),
        axis.title = element_text(colour = "black", face = "bold"),
        axis.text = element_text(colour = "black"),
        axis.ticks = element_line(colour = "black", size = 0.5)) +
  theme(plot.title = element_text(size = 20, face = "bold", hjust = 0.5)) +
  annotate("text", x = "5", y = 1.2,
            hjust = 1.2, vjust = -0.2, size = 6, colour = "dodgerblue",
            label = table(subset(sar.short, sar.short$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,
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) +

#box.short.slope

Figure 4d
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
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**

```r
# 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

```r
# 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**

```r
# 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