ERRATUM

Erratum to: Using Reflectance Ratios to Study Animal Coloration

Gonalo C. Cardoso¹ · Ana Cristina R. Gomes¹

Published online: 22 September 2015
© Springer Science+Business Media New York 2015

Erratum to: *Evol Biol* (2015) 42:387–394 DOI 10.1007/s11692-015-9328-5

The original version of this article unfortunately contained an error in Fig. 4 and the corresponding Appendix 3. The error is corrected with this erratum.

Appendix 3

We used 8911 reflectance spectrophotometry measurements of plumage colours, made on museum skins of 135 species of finches in the family Estrildidae. For most species male and female specimens were measured, usually three specimens per species/sex, and multiple measurements of colour were made that include all the main body parts. This dataset will be described in detail elsewhere.

Here we used visual models to estimate colour saturation of these spectra as the distance to the achromatic centre of an avian tetrahedral colour space (Endler and Mielke 2005; Stoddard and Prum 2008), either applying or not applying Fechner law. Reflectance values lower than 1 % were flattened to 1 %, and spectra were then analysed using the avian visual model of Vorobyev et al. (1998) as implemented in *pavo* (functions *vismodel* and *tcs*; Maia et al. 2013). Settings for the visual model without

Fechner law (y-axes in Fig. 4) were as follows: average UV avian sensitivity, regular daylight illumination, von Kries correction, relative quantum catches, no Fechner law (i.e. not log-transforming quantum catches), and the remaining options set to default.

For visual models with Fechner law (x-axes in Fig. 4) we first ran a model similar to the above but with Fechner law applied (i.e. log-transforming quantum catches), and with absolute rather than relative quantum catches. Then, as explained below, we rescaled log-transformed quantum catches to a biologically interpretable scale before normalization (i.e. dividing the signal of each photoreceptor cone type by the sum of signals of the four cone types) to obtain relative quantum catches.

Rescaling log-transformed quantum catches is necessary because, as discussed by Stoddard and Prum (2008; page 4 of the appendix), log-transformation changes colour spaces artefactually in a way that strongly depends on the arbitrary, unitless scale chosen for absolute quantum catches (from 0 for no stimulation of the photoreceptor, to an arbitrary value for stimulation by 100 % reflectance). Relative differences among log-transformed quantum catches are unaffected by the scale used or by the logarithmic base chosen, but the absolute values change so that normalizing log-quantum catches directly produces meaningless colour spaces. For example, software for analysis of spectral colour usually quantify quantum catches from 0 to an arbitrary value of 1; the resulting log-transformed quantum catches have negative values that, upon normalization, distort colour to the point that the relative strength of signals from the four cone types is reversed. The conceptual problem is that log-transformed quantum catches have no intrinsic minimum (as $\log(0)$ tends to minus infinity, though low values should be flattened because they reach beyond the perceptual range of animals and because

The online version of the original article can be found under doi:[10.1007/s11692-015-9328-5](https://doi.org/10.1007/s11692-015-9328-5).

✉ Gonalo C. Cardoso
gcardoso@cibio.up.pt

¹ CIBIO—Centro de Investigao em Biodiversidade e Recursos Genticos, Universidade do Porto, Campus Agrrio de Vairo, 4485–661 Vairo, Portugal

Fig. 4 Scatterplots of colour saturation, r (quantified as the distance to the achromatic centre of an avian colour space; Endler and Mielke 2005; Stoddard and Prum 2008), for ca. 9000 measurements of plumage colours across 135 species of estrildid finches, using visual models that do or do not apply Fechner law (i.e. log-transforming quantum catches of photoreceptor cells). Colour spaces that use

Fechner law are critically affected by the scale chosen (Stoddard and Prum 2008); **a**, **b** use rescaling of quantum catches that, respectively, likely under- and over-estimate the real similarity between colour spaces with and without Fechner law. See “Appendix 3” for details

of measurement inaccuracy; see ‘Recommendations’ in the main text); therefore normalization is meaningless unless log-transformed quantum catches are first rescaled for zero to be biologically sensible.

We rescaled log-transformed quantum catches as follows. The upper value, corresponding to stimulation by 100 % reflectance, was set to an arbitrary positive value: 1. The lower value in this dataset corresponds to stimulation by 1 % reflectance (since spectra are flattened at 1 %) and should be placed between 0 and 1, at a value that depicts the perceptual salience of variation below 1 % reflectance relative to the animals’ perceptual range. This value can be investigated empirically (e.g. Griebel and Schmid 1999; Scholtyssek et al. 2008). For the examples in Fig. 4 we chose two values that likely under- and over-estimate the perceptual salience of variation in reflectance below 1 %: panel A rescaled log-quantum catches between 0.01 (i.e. 1 %) and 1, which likely underestimates the perceptual

salience of reflectance under 1 % because animals should generally discriminate better among low than among high reflectance stimuli; panel B rescaled log-quantum catches between 0.2 (i.e. 20 %) and 1, which assumes a very large perceptual salience for reflectance under 1 %. For both panels, we first scaled between 0 and 1 by subtracting the minimum (i.e. log-transformed quantum catches of a flat spectrum at 1 % reflectance) and dividing by the range (i.e. log-transformed quantum catches of a flat spectrum at 100 % reflectance, minus log-transformed quantum catches of a flat spectrum at 1 % reflectance); then, for panel A we multiplied by 0.99 and added 0.01, to scale between 0.01 and 1, and for panel B multiplied by 0.8 and added 0.2, to scale between 0.2 and 1. Computing colour saturation by reference to tetrahedral colour spaces proceeded as usual (normalizing by dividing the signal of each cone type by the sum of signals of the four cone types, and then using the function `tcs` in `pavo`).