Marta Vidal-García | Lashi Bandara | J. Scott Keogh

ShapeRotator

An R tool for standardized rigid rotations of articulated three-dimensional structures with application for geometric morphometrics

ORIGINAL RESEARCH

ShapeRotator: An R tool for standardized rigid rotations of articulated three-dimensional structures with application for geometric morphometrics

Marta Vidal-García[1] [iD] | Lashi Bandara[2] | J. Scott Keogh[1]

[1]Ecology & Evolution, Research School of Biology, The Australian National University, Canberra, ACT, Australia

[2]Institut für Mathematik, Universität Potsdam, Potsdam OT Golm, Germany

Correspondence
Marta Vidal-García, Ecology & Evolution, Research School of Biology, The Australian National University, Canberra, ACT, Australia.
Email: marta.vidalga@gmail.com

Abstract

The quantification of complex morphological patterns typically involves comprehensive shape and size analyses, usually obtained by gathering morphological data from all the structures that capture the phenotypic diversity of an organism or object. Articulated structures are a critical component of overall phenotypic diversity, but data gathered from these structures are difficult to incorporate into modern analyses because of the complexities associated with jointly quantifying 3D shape in multiple structures. While there are existing methods for analyzing shape variation in articulated structures in two-dimensional (2D) space, these methods do not work in 3D, a rapidly growing area of capability and research. Here, we describe a simple geometric rigid rotation approach that removes the effect of random translation and rotation, enabling the morphological analysis of 3D articulated structures. Our method is based on Cartesian coordinates in 3D space, so it can be applied to any morphometric problem that also uses 3D coordinates (e.g., spherical harmonics). We demonstrate the method by applying it to a landmark-based dataset for analyzing shape variation using geometric morphometrics. We have developed an R tool (ShapeRotator) so that the method can be easily implemented in the commonly used R package *geomorph* and *MorphoJ* software. This method will be a valuable tool for 3D morphological analyses in articulated structures by allowing an exhaustive examination of shape and size diversity.

KEYWORDS
articulation, morphology, motion correction, multi-modular morphology

1 | BACKGROUND

Data on shape and size variation are essential in many fields, including evolutionary biology and ecology, engineering, medical science, and anthropology (Loncaric, 1998; McIntyre & Mossey, 2003; Slice, 2006). For most of these studies, the most widely used tools for analyzing morphological variation within or between a group of organisms or objects are based on Cartesian coordinates of landmarks (Bookstein, 1997).

Of the wide array of methods using Cartesian coordinates, geometric morphometrics (GM) is the most common, especially when analyzing shape and size variation and covariation (Adams, Rohlf, & Slice, 2013; Mitteroecker & Gunz, 2009). The first two steps of this GM procedure consist of a landmark approach that (1) gathers

(two- or three-dimensional) coordinates of anatomically defined and homologous loci, followed by (2) a generalized Procrustes analysis (GPA) that superimposes configurations of each set of landmarks in all specimens, by removing all effects of size, translation, and rotation, in order to only obtain shape information (Adams et al., 2013; Klingenberg, 2008). Geometric morphometrics, therefore, allows accurate quantitative analyses of shape and size, in either two-dimensional (2D) or three-dimensional (3D) space.

3D morphological analyses are the most accurate, as objects and organisms exist in 3D space. The recent growth in X-ray micro-CT scanning and surface scanning has seen a rapid increase in the application of 3D geometric morphometric techniques, but progress has been hampered by the lack of a simple method to incorporate data from complex articulated structures.

In evolutionary biology, identifying morphological differences among different groups or taxa is crucial in order to understand evolutionary processes and their relationship to the environment (Losos, 1990; Pagel, 1999; Ricklefs & Miles, 1994). This can be difficult, especially, if traits have coevolved, or if morphological diversification has been hindered by phylogenetic legacy or trade-offs imposed by the organism's functional habitat (Ghalambor, McKay, Carroll, & Reznick, 2007). Complex body shape patterns require more detailed analyses of shape, obtained by collecting data from several structures that capture the whole gamut of morphological variation in an organism. One example of this is the extraction and assembly of data from articulated structures, such as skeletons, for 3D analyses with geometric morphometric techniques. This is especially important in functional morphological studies, as they usually involve analyzing more than one structure due to mechanical correlations or morphological integration. For example, jointly analyzing the skull and mandible could be crucial to disentangle the relationship between diet

and head shape evolution (Cornette, Baylac, Souter, & Herrel, 2013). Similarly, collectively evaluating different modules in the limbs, especially when correlated to locomotion, or considering several structures across the whole body, could improve our understanding of the effect of environmental conditions on morphological evolution (Vidal-García & Keogh, 2017).

Unfortunately, nonrigid structures, such as articulated structures, will inevitably suffer the effects of natural or free rotation or translation events and be different in each individual and structure (Adams, 1999). These events could obstruct the correct quantification of shape variation by adding rotation artifacts to GM analyses (Adams, Rohlf, & Slice, 2004). Thus, orientation of these structures needs to be corrected and standardized prior to performing shape analyses. Several approaches for shape analysis of landmark data in articulated structures already have previously been proposed (Adams, 1999; Bookstein, 1991). However, even though these approaches can be used in 2D and also in 3D data with further modifications, currently, it has only been implemented in the two-dimensional space, such as in _geomorph_'s function fixed.angle() (Adams et al., 2017).

Here, we present the R tool _ShapeRotator_: a simple geometric rigid rotation approach to study three-dimensional (3D) shape of articulated structures, or independent structures, within an organism. We describe a method that removes shape variation due to the effect of translation between independent structures and rotation generated by movement in an articulation, among others. Thus, our approach translates and rotates articulated (or even independent) structures in order to obtain a comparable dataset with all effects of random movement and rotations removed.

We apply this method to two landmark-based datasets for analyzing shape variation using geometric morphometrics: (A) two

simple.rotation()
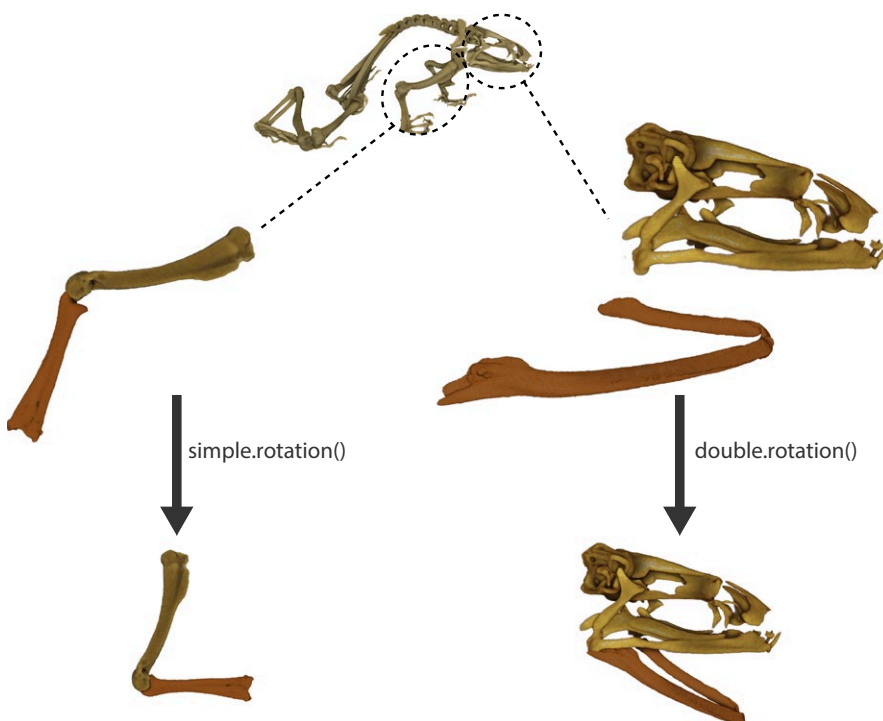
double.rotation()

FIGURE 1 Two structures in a simple-point articulation and a double-point articulation being rotated using _simple.rotation()_ and _double.rotation()_, respectively. The structures depicted are a frog humerus and radioulna for the single-point articulation and a skull and a detached mandible for the double-point articulation
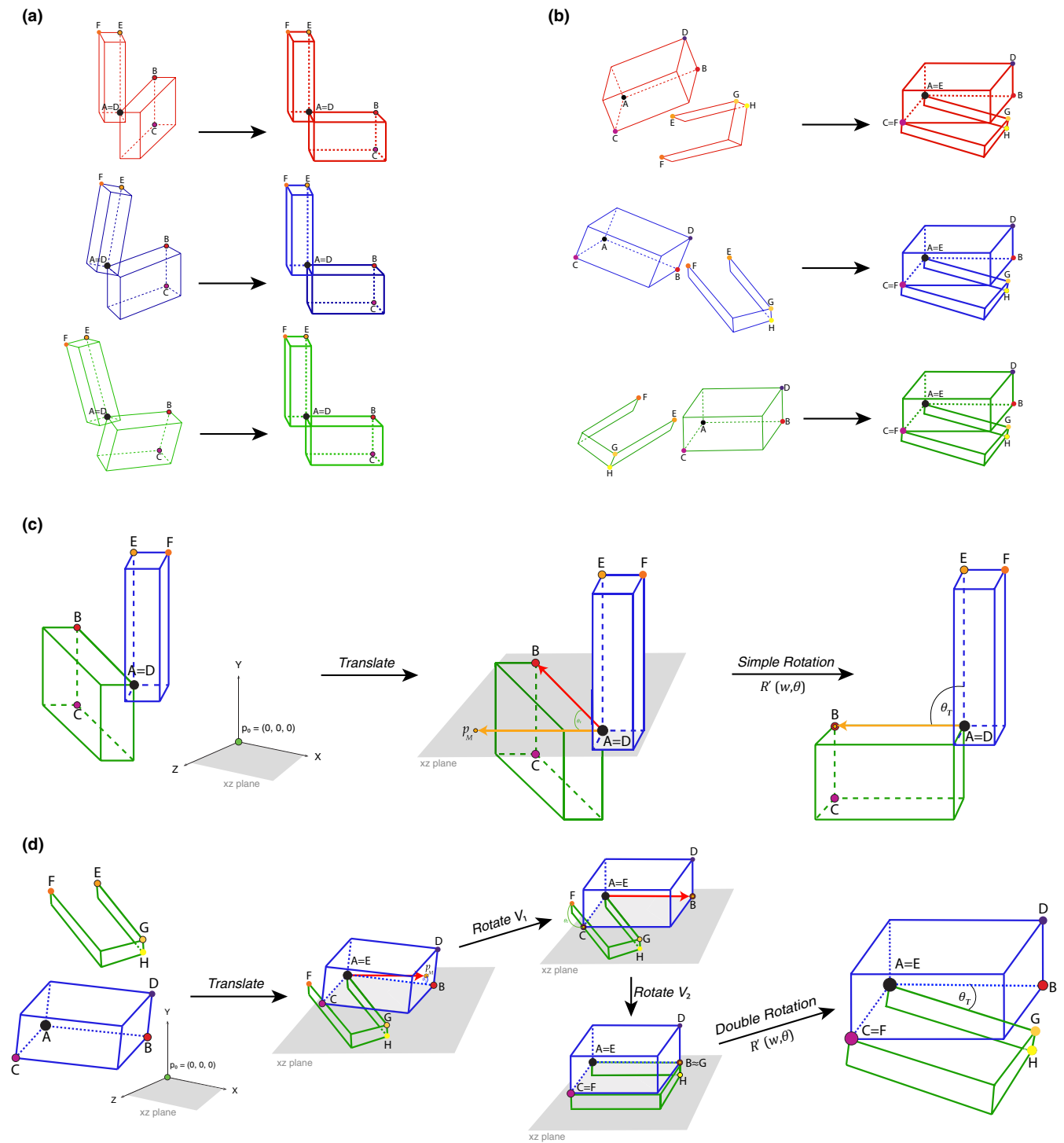
FIGURE 2 (a) Application of a translation -*translate()*- and the 3D rigid rotation method *simple.rotation()* in a simple-point articulation (e.g., humerus and radioulna) for three different "specimens," by rotating articulated structures to a standardized position relative to each other. (b) Application of the translation -*translate()*- and the 3D rigid rotation method *double.rotation()* in a double-point articulation for three different "specimens" (e.g., skull and mandibles). (c) Rotation method exemplified for *simple.rotation()* by depicting the plane spanned by the already translated point p_0 and A. Please note that p_0 depicts the origin point (0, 0, 0). The rotated resulting point p_M, landmarks B, C, D, E, and F, and angle θ_T (desired angle between the two structures) are also depicted. (d) Rotation method exemplified for *double.rotation()*. Please note that as both structures are not attached for the double-point articulation example, they are both translated to p_0, which depicts the origin point (0, 0, 0). The rotated resulting point p_M, landmarks A, B, C, D, E, F, G, and H, and angles θ_1, θ_2, and θ_T (desired angle between the two structures) are also depicted

substructures in a single-point articulation and (B) two substructures in a double-point articulation (Figures 1 and 2a,b). We also provide the example dataset used in *ShapeRotator* (available in GitHub) to execute these two kinds of rigid rotations, which then allows geometric morphometric analyses to be performed in the two most commonly used 3D GM analytical software packages: *geomorph* (Adams et al., 2017) and *MorphoJ* (Klingenberg, 2011). This method also will allow exporting of the rotated coordinates for posterior analyses in other software platforms, even outside of the field of geometric morphometrics. As the basis of this method lies upon rigidly spinning any structure defined by 3D coordinates, it could be used in any other shape analyses that use coordinate data, such as continuous surface meshes used in spherical harmonics (Shen, Farid, & McPeek, 2009). Our method is a convenient addition to the rapidly evolving tool kit of geometric morphometrics because it allows a more comprehensive exploration of

Let us describe this setup slightly more generally. For two vectors $u,v \in \mathbb{R}^3$, the axis to the plane spanned by these two vectors is determined by a unit normal to the plane (there are two choices due to orientation), which we denote by $N(u,v)$:

$$N(u,v) = \frac{u \times v}{|u \times v|} \qquad (2)$$

where $\times$ is the *cross product*. The angle between these vectors is then given by $\angle(u,v)$:

$$\angle(u,v) = \arccos\left(\frac{u \cdot v}{|u||v|}\right) \qquad (3)$$

where $\cdot$ is the *dot (scalar) product* between vectors. The rotation matrix about an axis $w \in \mathbb{R}^3$, where $w = (w_1, w_2, w_3)$ is a unit vector, of angle θ radians is given by the well-known matrix:

$$R'(w,\theta) = \begin{pmatrix} \cos\theta + w_1^2(1-\cos\theta) & w_1 w_2(1-\cos\theta) - w_3\sin\theta & w_1 w_3(1-\cos\theta) + w_2\sin\theta \\ w_2 w_1(1-\cos\theta) + w_3\sin\theta & \cos\theta + w_2^2(1-\cos\theta) & w_2 w_3(1-\cos\theta) - w_1\sin\theta \\ w_3 w_1(1-\cos\theta) - w_2\sin\theta & w_3 w_2(1-\cos\theta) + w_1\sin\theta & \cos\theta + w_3^2(1-\cos\theta) \end{pmatrix} \qquad (4)$$

morphological diversity through the gathering of shape data from complex 3D structures.

2 | METHODOLOGY

We begin with a set of points $\tilde{P} = \{\tilde{p}_0, \ldots, \tilde{p}_M\} \subset \mathbb{R}^3$. which represents a 3D object and are ordered so that $\tilde{p}_0$ represents the base point and $\tilde{p}_M$ represents the end point, by which we mean that this object has an axis starting from $\tilde{p}_0$ and ending at $\tilde{p}_M$. Our goal was to rotate these points via a rigid motion so that the axis on which these two points sit is either on the *X*, *Y*, or *Z*-axis in $\mathbb{R}^3$. Rotation of vectors in $\mathbb{R}^3$ is a well-known and easily resolved problem, and various formalisms exist in geometry. Thus, we translate our set of points $\tilde{P}$ so that $\tilde{p}_0$ maps to the origin (0, 0, 0). This is a simple transformation T defined by:

$$p_i = T\tilde{p}_i = \tilde{p}_i - \tilde{p}_0 \qquad (1)$$

Note that the axis X = span $\{(1, 0, 0)\}$, Y = span $\{(0, 1, 0)\}$, Z = span $\{(0, 0, 1)\}$, where each of the generating vectors is unit. Let us fix our desired axis to which we rotate the object to be A = span $\{a\}$ where $a = (1, 0, 0)$, or $a = (0, 1, 0)$, or $a = (0, 0, 1)$. As we have translated points $\{p_i\}$ and vectors correspond to positions, we are simply looking to rotate the vector p_M to A, and each other vector as a rigid motion with respect to this rotation. There are a number of ways to do this, but the simplest way is to consider the plane spanned by p_M and A and then to rotate by the angle between p_M and A within this plane (Figure 2c,d). Such a rotation is performed via rotating on the axis to the plane, which is determined by a normal vector to this plane.

Thus, to obtain a rotation matrix which is the rigid motion rotating the vector u to v in the plane spanned by u and v, we obtain the expression:

$$R(u,v) = R'(N(u,v), \angle(u,v)) \qquad (5)$$

Getting back to our original problem, we set $v = p_M$ and $u = a$, and then we have the rotated points:

$$r_i = R(p_M,a)p_i = R'(N(p_M,a), \angle(p_M,a,)) \, p_i \qquad (6)$$

where $R(p_M, a) \, p_i$ is the action of the matrix $R(p_M, a)$ on the vector p_i.

It may be necessary to introduce a further constraint in the rotation. For instance, suppose $a = (0, 1, 0)$ and there is a point p_l, now rotated to r_l via the method we describe, which should lie in the *Y*-axis. That is, we need to further rotate r_l to a point $r'_l = (*, *, 0)$. To do this, we simply rotate in the axis a, by an angle $\theta_Y(r_l) = \arctan((r_l)_3 / (r_l)_1)$, where $r_l = ((r_l)_1, (r_l)_2, (r_l)_3)$. That is,

$$r'_i = R'(a, \theta_Y(r_l))r_i \qquad (7)$$

3 | IMPLEMENTATION

3.1 | Overview of the *ShapeRotator* package

Here, we illustrate the functions available within the *ShapeRotator* R tool and the basic steps required in order to successfully implement the rotation on a dataset of 3D coordinates. *ShapeRotator* allows the rigid rotation of sets of both landmarks and semilandmarks used in geometric morphometric analyses, enabling morphometric analyses of complex objects, articulated structures, or multiple parts within

an object or specimen. This tutorial uses two example datasets: (A) two neighboring bones of the arm in a frog (humerus and radioulna), representing a single-point articulation example and (B) a skull and a mandible, representing a two-point articulation example (Figure 1). The main steps required are as follows: (1) importing the data and fixating the rotation axes, (2) translating the whole dataset of co-ordinates or points so that the main selected point $\bar{p}_0 = (0,0,0)$, and (3) rotating the two structures to the desired angle (as outlined on Figure 2a,b).

3.2 | Importing a dataset

In the first example dataset (example A), we use two geometric mor-phometric datasets containing both landmarks and semilandmarks for two neighboring and articulated bones (humerus and radioulna) from a group of several species of frogs (details in Appendix S1), in tps format. We first import the datasets using the R package *geo-morph* (Adams et al., 2017):

```
library(devtools)
install _ github("marta-vidalgarcia/ShapeRotator")
library(geomorph)
radioulna <- readland.tps("radioulna.tps", specID = "ID",
readcurves = F)
humerus <- readland.tps("humerus.tps", specID = "ID",
readcurves = F)
```

These two GM datasets (radioulna and humerus) will be ro-tated on different rotation axes in order to conform the aimed angle between them. This process is not exclusive to two neigh-boring structures, and thus, it could be performed for as many independent subunits as desired by choosing the different angles among different rotation axes and all the translation processes. For more help on importing the GM datasets, please see Adams et al., (2017) and the associated help files. Please note that this method also works for semilandmarks as long as they have been equidistantly positioned prior to the translation and the rigid rotation. The same process is needed for the second example dataset (example B).

3.3 | Translating

During this step, each structure will be translated to the point of origin so that $\bar{p}_0 = (0,0,0)$; thus, the distance from the coordinates of landmark_a (A_x, A_y, A_z) is substracted from all the landmarks in all specimens, for example ($N_x - A_x$, $N_y - A_y$, $N_z - A_z$) for landmark N. This translation is made with the function *translate()*, as it follows for the one-point articulation example (example A):

```
data.1 _ t <- translate (T=data _ 1, landmark=landmarkA)
data.2 _ t <- translate (T=data _ 2, landmark=landmarkD)
```

And for the double-point articulation (example B):

```
data.3 _ t <- translate (T=data _ 3, landmark=landmarkA)
data.4 _ t <- translate (T=data _ 4, landmark=landmarkE)
```

Please note that the default is to set the origin point to, but this can be changed to another origin point. For example:

```
data.1 _ t  <-  translate (T=data _ 1, landmark=land-
markA, origin = c(1,3,5))
data.2 _ t  <-  translate (T=data _ 2, landmark=land-
markD, origin = c(1,3,5))
```

3.4 | Fixing the rotation axis

In order to fix a rotation axis in a structure, we first need to select in our dataset two suitable landmarks for each structure which the rotation axis will go through landmarks A and B (for the first struc-ture), and landmarks D and E (for the other structure). In the radi-oulna dataset from example A, landmark A is the landmark in the 1st position and landmark B is in the 10th position. Similarly, for the humerus dataset, landmark D would be the landmark on the 52nd position, and landmark E would be in the 19th position. Please note that in example A, we would perform a simple rotation between two structures, based on the angle formed by three points (one on each structure and the articulation point itself. However, as we aim to remove all articulation-related variation, including torsional ro-tations and mirroring artefacts, we need an extra landmark in each structure that shares at least one of their coordinates (but ideally two) with the distal landmark (e.g., $y_A = y_B$ for the first structure, and $z_D = z_E$ for the second). This extra landmark is needed for the simple reason that there is not enough information about the ori-entation of the structure with only two landmarks per structure, so even though the rigid rotation will work properly, it could position this structure in the wrong "mirroring" orientation (Figure 2c). This orientation issue is corrected by adding landmark C. In this exam-ple, landmark C is the 17th landmark, while landmark F is the 107th landmark in the humerus dataset. We need to know which land-marks will be selected in both structures prior to the rotation pro-cess in order to insure that the rotation process will work properly. Setting up landmarks that will be used in the rotation process of example B is very similar to example A, but it needs four landmarks for each structure instead of three, as it is a more complex articula-tion that requires two separate rotations per structure (Figure 2d). Even though each of these rotations are calculated internally (only the four landmarks per structure and the desired angle between them need to be provided), it will be beneficial to choose land-marks that are spatially arranged in a way that facilitates the ro-tation process, and results in the rotating multistructure being placed in a biologically relevant angle between each substructure. ShapeRotator will give a warning message if the landmarks chosen are not optimal.

3.5 | Rotating

In the rotation step, we will use the function *simple.rotation*() in order to rigidly rotate the two structures of example A to the desired angle (in degrees), as it follows:

```
joined _ dataset <- simple.rotation(data.1 = data.1 _ t,
data.2 = data.2 _ t, land.a = 10, land.b=1, land.c=17,
land.d=52, land.e=19, land.f=107, angle = 90)
```

The input datasets data.1 and data.2 correspond to the two translated datasets (in this case translated_radioulna and translated_humerus. We then use the selected landmarks as explained in the previous section. Finally, we include the angle (in degrees) that we would like to use to position the two structures to one another. One of the options of the function *simple.rotation*() is to select the desired angle between the two structures so that we can perform the rigid rotation of each structure positioning them in the selected angle in relation to each other. In order to do so, we use the internal function *vector.angle()*, by providing the desired angle in degrees (from 0° to 360°). The function *vector.angle()* will return angle.v, a vector that forms that angle with the vector (1, 0, 0). In the example dataset in ShapeRotator, we rotate the coordinates from the two bones so that they form an angle of 90° degrees to each other:

```
New_vector <- vector.angle(90)
```

So that New_vector = c(0, 1, 0). Thus, we could check the vector that the function *simple.rotation*() will use, based on the input angle. The output from the function *simple.rotation*() is a 3D array with the two joined datasets (data.1 and data.2). Please note that the two datasets are joined based on their associated specimen names (dimnames). Thus, the order of the specimens in each dataset is not important, as long as all the cases match perfectly between the two datasets in the same specimen. If there are extra specimens for one of the datasets or the names do not match properly, *simple.rotation*() will not include them in the output rotated joined dataset.

As explained in the previous section, the function used for the rotation step for example B needs four landmarks for each structure instead of three, as it is a more complex articulation that will need to get landmarks from both structures joined (or very close) to form the double-articulation hinges, as well as internally running separate rotations per structure on top of the final rotation. We will use *double.rotation*() function. For this function, we will need to match the dimnames:

```
matched = match.datasets(data.3_t, data.4_t)
```

The datasets that will need to be rotated are the following:

```
data.3_rotate = matched$matched1
data.4_rotate = matched$matched2
```

Similarly to *simple.rotation()*, the *double.rotation()* will be as follows:

```
dr45l=double.rotation(data.1=data.3_rotate,data.2=
data.4_rotate,land.a=1,land.b=2,land.c=3,land.d=4,
land.e=1, land.f=2, land.g=3, land.h=4, angle=45)
joined_dataset <- double.rotation(data.1 = trans-
lated_skull, data.2 = translated_mandible, land.a
= 1, land.b=2, land.c=3, land.d=4, land.e=1, land.
f=2, land.g=3, land.h=4, angle = 45)
```

The function double.rotation() will return a list that will need to be joined as it follows:

```
dr45 = join.arrays (dr45l$rotated1, dr45l$rotated2)
```

Double.rotation() allows the user to translate one of the objects after the rotation, in the case of not wanting them in contact to one another. For example

```
skull_translate = c(1.7,0.1, 0)
```

```
dr45_st = join.arrays (dr45l$rotated1, trans-
late(dr45l$rotated2,land.e, skull_translate))
```

3.6 | Plotting

Finally, after the rotation process, we can visualize a 3D plot for each specimen with the two rotated structures for both a single-point and a double-point articulations using the function *plot.rotation.3D()*:

```
plot.rotation.3D(joined.data=joined_dataset,data.1=
data.1, data.2=data.2, specimen.num =1)
plot.rotation.3D(joined.data=dr45_st,data.1=data.3,
data.2=data.4, specimen.num =1)
```

The default plotting colors for the landmarks of the joined rotated structure (joined.data) is black and red for each original structure (data.1 and data.2).

3.7 | Exporting

After the rotation process, we could either use the joined GM array in further analyses, visualize the resulting joined structure through *geomorph* (Adams et al., 2017), or we could also export it and save it in order to use it in another software, such as MorphoJ (Klingenberg, 2011). In this step, we will be using the function writeland.tps() in the R package *geomorph* (Adams et al., 2017) in order to save a tps file from the joined GM array:

```
writeland.tps(A="joined_arm", file = "joined_arm.
tps", scale = NULL)
```

3.8 | Other applications

Our method is an important addition to the tool kit of the geometric morphometrics field. It will facilitate the analyses of compound 3D morphological datasets in geometric morphometrics analyses but will also be useful outside of this field as it can be applied to any method that uses 3D coordinates. The examples of applications are numerous in different fields of study, such as biology, anthropology, paleontology, medical sciences, archeology, and engineering. For example, in evolutionary biology, ShapeRotator would allow analyses of multiple or articulated hard structures (such as different segments of an exoskeleton, different articulated bones, or neighboring plant structures, among others), different structures from the same object or organism (e.g., different and not adjacent body parts), or pieces from damaged specimens. In medicine and veterinary science, it could be used to examine shape and size variation in different organisms' growth due to different nutritional treatments or to examine how different structures respond to injuries or surgery. It would be useful in paleontology or archeology when trying to quantify shape of different objects or organisms that might have been preserved in disarticulated pieces. Finally, we would like to include a word of caution on how the angle chosen between different substructures could produce different results. As common sense would suggest, the angle chosen should always be within the range of angles possible for that complex object, but in some cases, it might be necessary

to perform the GM analysis on more than one angle to be sure that the results do not differ. In other cases, it might even be better to analyse substructures separately.

CONFLICT OF INTEREST

None declared.

ORCID

Marta Vidal-García _{iD} http://orcid.org/0000-0001-7617-7329

REFERENCES

Adams, D. C. (1999). Methods for shape analysis of landmark data from articulated structures. _Evolutionary Ecology Research_, _1_, 959–970.

Adams, D., Collyer, E., Kaliontzopoulou, A., Sherratt, E. (2017). geomorph: An R package for the collection and analysis of geometric morphometric shape data. _Methods in Ecology and Evolution_, _4_, 393–399. https://doi.org/10.1111/2041-210X.12035

Adams, D. C., Rohlf, F. J., & Slice, D. E. (2004). Geometric morphometrics: Ten years of progress following the 'revolution'. _Italian Journal of Zoology_, _71_, 5–16. https://doi.org/10.1080/11250000409356545

Adams, D. C., Rohlf, F. J., & Slice, D. E. (2013). A field comes of age: Geometric morphometrics in the 21st century. _Hystrix_, _the Italian Journal of Mammalogy_, _24_, 7–14.

Bookstein, F. L. (1991). _Morphometric tools for landmark data: Geometry and biology_ (p. 435). New York, NY: Cambridge University Press.

Bookstein, F. L. (1997). Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape. _Medical Image Analysis_, _1_, 225–243. https://doi.org/10.1016/S1361-8415(97)85012-8

Cornette, R., Baylac, M., Souter, T., & Herrel, A. (2013). Does shape covariation between the skull and the mandible have functional consequences? A 3D approach for a 3D problem. _Journal of Anatomy_, _223_, 329–336. https://doi.org/10.1111/joa.12086

Ghalambor, C. K., McKay, J. K., Carroll, S. P., & Reznick, D. N. (2007). Adaptive versus non-adaptive phenotypic plasticity and the potential

for contemporary adaptation in new environments. _Functional Ecology_, _21_, 394–407. https://doi.org/10.1111/j.1365-2435.2007.01283.x

Klingenberg, C. P. (2008). Morphological integration and developmental modularity. _Annual Review of Ecology, Evolution, and Systematics_, _39_, 115–132. https://doi.org/10.1146/annurev.ecolsys.37.091305.110054

Klingenberg, C. P. (2011). MorphoJ: An integrated software package for geometric morphometrics. _Molecular Ecology Resources_, _11_, 353–357. https://doi.org/10.1111/j.1755-0998.2010.02924.x

Loncaric, S. (1998). A survey of shape analysis techniques. _Pattern Recognition_, _31_, 983–1001. https://doi.org/10.1016/S0031-2023(97)00122-2

Losos, J. B. (1990). The evolution of form and function: Morphology and locomotor performance in West Indian Anolis lizards. _Evolution_, _44_, 1189. https://doi.org/10.1111/j.1558-5646.1990.tb05225.x

McIntyre, G. T., & Mossey, P. A. (2003). Size and shape measurement in contemporary cephalometrics. _The European Journal of Orthodontics_, _25_, 231–242. https://doi.org/10.1093/ejo/25.3.231

Mitteroecker, P., & Gunz, P. (2009). Advances in geometric morphometrics. _Evolutionary Biology_, _36_, 235–247. https://doi.org/10.1007/s11692-009-9055-x

Pagel, M. (1999). Inferring the historical patterns of biological evolution. _Nature_, _401_, 877–884. https://doi.org/10.1038/44766

Ricklefs, R. E., & Miles, D. B. (1994). Ecological and evolutionary inferences from morphology: An ecological perspective. In P. C. Wainwright, & S. M. Reilly (Eds.), _Ecological morphology: Integrative organismal biology_ (pp. 13–41). Chicago, IL: University of Chicago Press.

Shen, L., Farid, H., & McPeek, M. A. (2009). Modeling three-dimensional morphological structures using spherical harmonics. _Evolution_, _63_, 1003–1016. https://doi.org/10.1111/j.1558-5646.2008.00557.x

Slice, D. E. (2006). _Modern morphometrics in physical anthropology_, Vol. _6n_. New York, NY: Springer.

Vidal-García, M., & Keogh, J. S. (2017). Phylogenetic conservatism in skulls and evolutionary lability in limbs–morphological evolution across an ancient frog radiation is shaped by diet, locomotion and burrowing. _BMC Evolutionary Biology_, _17_, 165. https://doi.org/10.1186/s12862-017-0993-0

SUPPORTING INFORMATION

Additional Supporting Information may be found online in the supporting information tab for this article.