

1 **ShapeRotator: an R package for standardised rigid rotations of articulated Three-Dimensional**
2 **structures with application for geometric morphometrics**

3

4

5 Marta Vidal-García^{1,*}, Lashi Bandara², and J. Scott Keogh¹

6

7

8

9 Author information:

10 ¹*Evolution, Ecology & Genetics, Research School of Biology, The Australian National University,*
11 *Canberra, ACT, Australia*

12

13 ²*Department of Mathematical Sciences, Chalmers University of Technology and University of*
14 *Gothenburg, Gothenburg, Sweden*

15

16 *Correspondence: Marta Vidal-García; Evolution, Ecology & Genetics, Research School of Biology,
17 The Australian National University, Canberra ACT 2601; Australia

18 e-mail: marta.vidalgarcia@anu.edu.au

19

20 **SHORT RUNNING TITLE**

21 Rigid rotation of 3D structures

22 **Summary**

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

42

43 **Key words:** morphology – articulation – motion correction – multi-modular morphology

44 **Background**

45 Data on shape and size variation is essential in many fields, including evolutionary biology
46 and ecology, engineering, medical science, and anthropology (Loncaric 1998; McIntyre & Mossey
47 2003; Slice 2006). For most of these studies, one of the most widely used tools for analysing
48 morphological variation within or between a group of organisms or objects are based on Cartesian
49 coordinates of landmarks (Bookstein 1997).

50 Of the wide array of methods using Cartesian coordinates, geometric morphometrics (GM) is the
51 most common, especially when analysing shape and size variation and covariation (Mitteroecker &
52 Gunz 2009; Adams *et al.* 2013). The first two steps of this GM procedure consist of a landmark
53 approach that: (1) gathers (two- or three-dimensional) coordinates of anatomically defined and
54 homologous loci, followed by (2) a generalised Procrustes analysis (GPA) that superimposes
55 configurations of each set of landmarks in all specimens, by removing all effects of size, translation
56 and rotation, in order to only obtain shape information (Klingenberg 2008; Adams *et al.* 2013).
57 Geometric morphometrics, therefore, allows accurate quantitative analyses of shape and size, in either
58 Two-Dimensional (2D) or Three-Dimensional (3D) space.

59 3D morphological analyses are the most accurate, as objects and organisms exist in 3D space. The
60 recent growth in x-ray micro CT scanning and surface scanning has seen a rapid increase in the
61 application of 3D geometric morphometric techniques, but progress has been hampered by the lack of a
62 simple method to incorporate data from complex articulated structures.

63 In evolutionary biology, identifying morphological differences among different groups or taxa is
64 crucial in order to understand evolutionary processes and their relationship to the environment (Losos
65 1990; Ricklefs & Miles 1994; Pagel 1999). This can be difficult, especially if traits have co-evolved, or
66 if morphological diversification has been hindered by phylogenetic legacy or trade-offs imposed by the
67 organism's functional habitat (Ghalambor *et al.* 2007). Complex body shape patterns require more
68 detailed analyses of shape, obtained by collecting data from several structures that capture the whole
69 gamut of morphological variation in an organism. One example of this is the extraction and assembly
70 of data from articulated structures, such as skeletons, for 3D analyses with geometric morphometric
71 techniques. This is especially important in functional morphological studies, as they usually involve
72 analysing more than one structure due to mechanical correlations or morphological integration. For
73 example, jointly analysing skull and mandible could be crucial to disentangle the relationship between

74 diet and head shape evolution (Cornette *et al.* 2013). Similarly, collectively evaluating different
75 modules in the limbs, especially when correlated to locomotion, or considering several structures
76 across the whole body, could improve our understanding of the effect of environmental conditions on
77 morphological evolution (Vidal-Garcia & Keogh, unpublished).

78 Unfortunately, non-rigid structures, such as articulated structures, will inevitably suffer the effects
79 of natural or free rotation or translation events and be different in each individual and structure
80 (Adams, 1999). These events could obstruct the correct quantification of shape variation by adding
81 rotation artifacts to GM analyses (Adams *et al.* 2004). Thus, orientation of these structures needs to be
82 corrected and standardised prior to performing shape analyses. Methods for shape analysis of landmark
83 data in articulated structures already have been described but the solution to this problem has been
84 implemented only in two-dimensional (2D) space (Adams, 1999).

85 Here we present the R package ShapeRotator: a simple geometric rigid rotation approach to study
86 3-Dimensional (3D) shape of articulated structures, or independent structures, within an organism. We
87 describe a method that removes shape variation due to the effect of translation between independent
88 structures and rotation generated by movement in an articulation, among others. Thus, our approach
89 translates and rotates articulated (or even independent) structures in order to obtain a comparable shape
90 data set once all effects of random movement and rotations have been removed (Fig. 1a).

91 We apply this method to a landmark-based data set for analysing shape variation using geometric
92 morphometrics, and provide the example data set used in ShapeRotator (to be available in CRAN) to
93 execute this rigid rotation. This rigid rotation then allows geometric morphometric analyses to be
94 performed in the two best well-known 3D GM analytical software packages: *geomorph* (Adams &
95 Otárola-Castillo 2013), and *MorphoJ* (Klingenberg 2011). This method also will allow exporting the
96 rotated coordinates for posterior analyses in other software platforms, even outside of the field of
97 geometric morphometrics. Since the basis of this method lies upon rigidly spinning any structure
98 defined by 3D coordinates, it could be used in any other shape analyses that use coordinate data, such
99 as continuous surface meshes used in spherical harmonics (Shen *et al.* 2009). Our method is a
100 convenient addition to the rapidly evolving tool kit of geometric morphometrics because it allows a
101 more comprehensive exploration of morphological diversity through the gathering of shape data from
102 complex 3D structures.

103

104 Methodology

105 We begin with a set of points $\tilde{P} = \{\tilde{p}_0, \dots, \tilde{p}_M\} \subset \mathbb{R}^3$ which represents a 3D object, and are
 106 ordered so that \tilde{p}_0 represents the base point and \tilde{p}_M represents the end point, by which we mean that
 107 this object has an axis starting from \tilde{p}_0 and ending at \tilde{p}_M . Our goal is to rotate these points via a rigid
 108 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
 109 vectors in \mathbb{R}^3 is a well-known and easily resolved problem, and various formalisms exist in geometry.
 110 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
 111 transformation T defined by:

$$112 \quad (1) \quad p_i = T\tilde{p}_i = \tilde{p}_i - \tilde{p}_0$$

113 Note that the axis $X = \text{span}\{(1, 0, 0)\}$, $Y = \text{span}\{(0, 1, 0)\}$, $Z = \text{span}\{(0, 0, 1)\}$, where
 114 each of the generating vectors are unit. Let us fix our desired axis to which we rotate the object to be
 115 $A = \text{span}\{a\}$ where $a = (1, 0, 0)$, or $a = (0, 1, 0)$, or $a = (0, 0, 1)$. Since we have translated points
 116 $\{p_i\}$ and vectors correspond to positions, we are simply looking to rotate the vector p_M to A , and each
 117 other vector as a rigid motion with respect to this rotation. There are a number of ways to do this, but
 118 the simplest way is to consider the plane spanned by p_M and A , and then to rotate by the angle between
 119 p_M and A within this plane (Fig. 1b). Such a rotation is done via rotating on the axis to the plane, which
 120 is determined by a normal vector to this plane.

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

$$124 \quad (2) \quad N(u, v) = \frac{u \times v}{|u \times v|}$$

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

$$126 \quad (3) \quad \angle(u, v) = \arccos\left(\frac{u \cdot v}{|u||v|}\right)$$

127 where \cdot is the *dot (scalar) product* between vectors. The rotation matrix about an axis $w \in \mathbb{R}^3$, where
 128 $w = (w_1, w_2, w_3)$ is a unit vector, of angle θ radians is given by the well known matrix:

$$129 \quad (4) \quad R'(w, \theta) =$$

$$130 \quad \begin{pmatrix} \cos \theta + w_1^2(1 - \cos \theta) & w_1 w_2(1 - \cos \theta) - w_3 \sin \theta & w_1 w_3(1 - \cos \theta) + w_2 \sin \theta \\ w_2 w_1(1 - \cos \theta) + w_3 \sin \theta & \cos \theta + w_2^2(1 - \cos \theta) & w_2 w_3(1 - \cos \theta) - w_1 \sin \theta \\ w_3 w_1(1 - \cos \theta) - w_2 \sin \theta & w_3 w_2(1 - \cos \theta) + w_1 \sin \theta & \cos \theta + w_3^2(1 - \cos \theta) \end{pmatrix}$$

131

132 Thus, to obtain a rotation matrix which is the rigid motion rotating the vector u to

133 v in the plane spanned by u and v , we obtain the expression:

134 (5)
$$R(u, v) = R'(N(u, v), \angle(u, v))$$

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

136 (6)
$$r_i = R(p_M, a)p_i = R'(N(p_M, a), \angle(p_M, a))p_i$$

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

138 It may be necessary to introduce a further constraint in the rotation. For instance, suppose $a = (0, 1, 0)$

139 and there is a point p_i , now rotated to r_i via the method we describe, which should lie in the Y -axis.

140 That is, we need to further rotate r_i to a point $r'_i = (*, *, 0)$. To do this, we simply rotate in the axis a ,

141 by an angle $\theta_Y(r_i) = \arctan((r_i)_3 / (r_i)_1)$, where $r_i = ((r_i)_1, (r_i)_2, (r_i)_3)$. That is,

142 (7)
$$r'_i = R'(a, \theta_Y(r_i))r_i$$

143

144 **Implementation**

145 **Overview of the ShapeRotator package**

146 Here we illustrate the functions available within the ShapeRotator package and the basic steps
147 required in order to successfully implement the rotation on a data set of 3D coordinates. ShapeRotator
148 allows the rigid rotation of sets of both landmarks and semi-landmarks used in geometric
149 morphometric analyses, enabling morphometric analyses of complex objects, articulated structures, or
150 multiple parts within an object or specimen. The main steps required are: (1) importing the data and
151 fixating the rotation axes, (2) translating the whole data set of coordinates or points so that the main
152 selected point $\tilde{p}_0 = (0, 0, 0)$, and (3) rotating the two structures to the desired angle (as outlined on Fig.
153 2). This tutorial uses the example data set, which is included in this package.

154

155 **Importing a data set**

156 In the example data set we use two geometric morphometric data sets containing both
157 landmarks and semi-landmarks for two neighbouring and articulated bones (humerus and radioulna)
158 from a group of several species of frogs (details in Appendix S1), in tps format. We first import the
159 data sets using the R package *geomorph* (Adams & Otárola-Castillo 2013):

160 `library(ShapeRotator)`

161 `library(geomorph)`

162 radioulna <- readland.tps("radioulna.tps", specID = "ID", readcurves = F)

163 humerus <- readland.tps("humerus.tps", specID = "ID", readcurves = F)

164

165 These two GM data sets (radioulna and humerus) will be rotated on different rotation axes in
166 order to conform the aimed angle between them. This process is not exclusive to two neighbouring
167 structures, and thus, it could be performed for as many independent subunits as desired by choosing the
168 different angles among different rotation axes and all the translation processes. For more help on
169 importing the GM data sets please see Adams *et al.* (2014), and the associated help files. Please note
170 that this method also works for semi-landmarks as long as they have been equally-distance positioned
171 prior to the translation and the rigid rotation.

172

173 **Translating**

174 During this step each structure will be translated to the point of origin so that $\tilde{p}_0 = (0, 0, 0)$,
175 thus the distance from the coordinates of landmark_a (A_x, A_y, A_z) is subtracted from all the
176 landmarks in all specimens, e.g. ($N_x - A_x, N_y - A_y, N_z - A_z$) for landmark N. This translation is
177 made with the function *translate()*, as it follows:

178 translated_radioulna <- translate (radioulna[RU_landmark_a, , 1], radioulna)

179 translated_humerus <- translate (humerus[H_landmark_a, , 1], humerus)

180

181 **Fixing the rotation axis**

182 In order to fix a rotation axis in a structure we first need to select in our data set two suitable
183 landmarks for each structure which the axis will go through: landmarks A and B (for the first
184 structure), and landmarks D and E (for the other structure). In the radioulna example data set, landmark
185 A is the landmark in the 1st position and landmark B is in the 10th position. Similarly, for the humerus
186 data set, landmark D would be the landmark on the 52th position, and landmark E would be in the 19th
187 position. Finally, we also need to select an extra landmark that shares the same value for at least one of
188 its coordinates (two coordinates, ideally) with landmark b. This is needed for the simple reason that
189 there is not information about the orientation of the structure with only two landmarks per structure, so
190 even though the rigid rotation will work properly, it could position this structure in the wrong
191 ('mirroring') orientation. Thus, this orientation issue is corrected through landmark C. In this example,

192 landmark C is the 17th landmark, while landmark F is the 107th landmark in the humerus data set. We
193 need to know which landmarks will be selected in both structures prior to the rotation process.

194

195

196 **Rotating**

197 In the rotation step, we will use the function `rotation()` in order to rigidly rotate the two
198 structures to the desired angle, as it follows:

```
199 joined_dataset <- rotation(data.1 = translated_radioulna, data.2 = translated_humerus, land.a =  
200 10, land.b=1, land.c=17, land.d=52, land.e=19, land.f=107, angle = 90)
```

201

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

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

213 So that `New_vector = c(0, 1, 0)`. Thus we could check the vector that the function `rotator()` will
214 use, based on the input angle. The output from the function `rotator()` is a 3D array with the two joined
215 data sets (`data.1` and `data.2`). Please note that the two datasets are joined based on their dimnames.

216 Thus, the order of the specimens in each dataset is not important, as long as all the cases match
217 perfectly between the two datasets in the same specimen. If there are extra specimens for one of the
218 datasets or the names do not match properly, `rotator()` will not include them in the output rotated joined
219 dataset.

220

221 **Exporting**

222 After the rotation process we could either use the joined GM array in further analyses or we
223 could also export it and save it in order to use it in another software, such as MorphoJ (Klingenberg
224 2011). In this step we will be using the function `writeland.tps()` in the R package *geomorph* (Adams &
225 Otárola-Castillo 2013) in order to save a tps file from the joined GM array:

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

227

228 **Other applications**

229 Our method is an important addition to the tool kit of the geometric morphometrics field. It will
230 facilitate the analyses of compound 3D morphological datasets in geometric morphometrics analyses
231 but will also be useful outside of this field as it can be applied to any method that uses 3D coordinates.
232 The examples of applications are numerous in different fields of study, such as biology, anthropology,
233 palaeontology, medical sciences, archaeology, and engineering. For example, in evolutionary biology,
234 ShapeRotator would allow analyses of multiple or articulated hard structures (such as different
235 segments of an exoskeleton, different articulated bones, or neighbouring plant structures, among
236 others), different structures from the same object or organism (e.g. different and not adjacent body
237 parts), or pieces from damaged specimens. In medicine and veterinary science it could be used to
238 examine shape and size variation in different organisms' growth due to different nutritional treatments
239 or to examine how different structures respond to injuries or surgery. It would be useful in
240 palaeontology or archaeology when trying to quantify shape of different objects or organisms that
241 might have been preserved in disarticulated pieces.

242

243 **Acknowledgements**

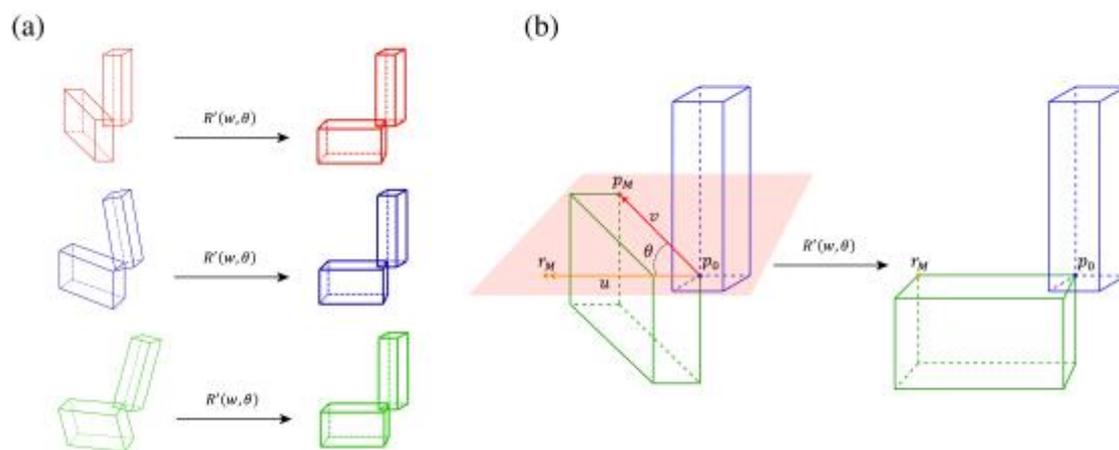
244 We are grateful to A. K. Singh and L. Bailey for their helpful comments on the manuscript. JSK thanks
245 the Australian Research Council for ongoing support.

246 **REFERENCES**

- 247 Adams, D.C. & Otárola-Castillo, E. (2013). geomorph: an R package for the collection and analysis of
248 geometric morphometric shape data. *Methods in Ecology and Evolution*, **4**, 393–399.
- 249 Adams, D.C., Rohlf, F.J. & Slice, D.E. (2013). A field comes of age: geometric morphometrics in the
250 21st century. *Hystrix, the Italian Journal of Mammalogy*, **24**, 7–14.
- 251 Adams, D.C., Rohlf, F.J. & Slice, D.E. (2004). Geometric morphometrics: Ten years of progress
252 following the ‘revolution’. *Italian Journal of Zoology*, **71**, 5–16.
- 253 Adams, D.C. (1999). Methods for shape analysis of landmark data from articulated structures.
254 *Evolutionary Ecology Research*, **1**, 959–970.
- 255 Bookstein, F.L. (1997). Landmark methods for forms without landmarks: morphometrics of group
256 differences in outline shape. *Medical Image Analysis*, **1**, 225–243.
- 257 Cornette, R., Baylac, M., Souter, T. & Herrel, A. (2013). Does shape co-variation between the skull
258 and the mandible have functional consequences? A 3D approach for a 3D problem. *Journal of*
259 *anatomy*, **223**, 329–36.
- 260 Ghalambor, C.K., McKay, J.K., Carroll, S.P. & Reznick, D.N. (2007). Adaptive versus non-adaptive
261 phenotypic plasticity and the potential for contemporary adaptation in new environments.
262 *Functional Ecology*, **21**, 394–407.
- 263 Klingenberg, C.P. (2011). MorphoJ: an integrated software package for geometric morphometrics.
264 *Molecular ecology resources*, **11**, 353–357.
- 265 Klingenberg, C.P. (2008). Morphological Integration and Developmental Modularity. *Annual Review*
266 *of Ecology, Evolution, and Systematics*, **39**, 115–132.
- 267 Loncaric, S. (1998). A survey of shape analysis techniques. *Pattern Recognition*, **31**, 983–1001.
- 268 Losos, J.B. (1990). The Evolution of Form and Function: Morphology and Locomotor Performance in
269 West Indian Anolis Lizards. *Evolution*, **44**, 1189.
- 270 McIntyre, G.T. & Mossey, P.A. (2003). Size and shape measurement in contemporary cephalometrics.
271 *The European Journal of Orthodontics*, **25**, 231–242.
- 272 Mitteroecker, P. & Gunz, P. (2009). Advances in Geometric Morphometrics. *Evolutionary Biology*, **36**,
273 235–247.
- 274 Pagel, M. (1999). Inferring the historical patterns of biological evolution. *Nature*, **401**, 877–84.
- 275 Ricklefs, R.E. & Miles, D.B. (1994). Ecological and evolutionary inferences from morphology: an
276 ecological perspective. In: *Ecological morphology: integrative organismal biology.*, pp. 13–41.
277 University of Chicago Press, Chicago.
- 278 Shen, L., Farid, H. & McPeck, M.A. (2009). Modeling three-dimensional morphological structures
279 using spherical harmonics. *Evolution*, **63**, 1003–16.
- 280 Slice, D.E. (2006). *Modern Morphometrics in Physical Anthropology*, Vol. 6n. Springer Science &
281 Business Media.

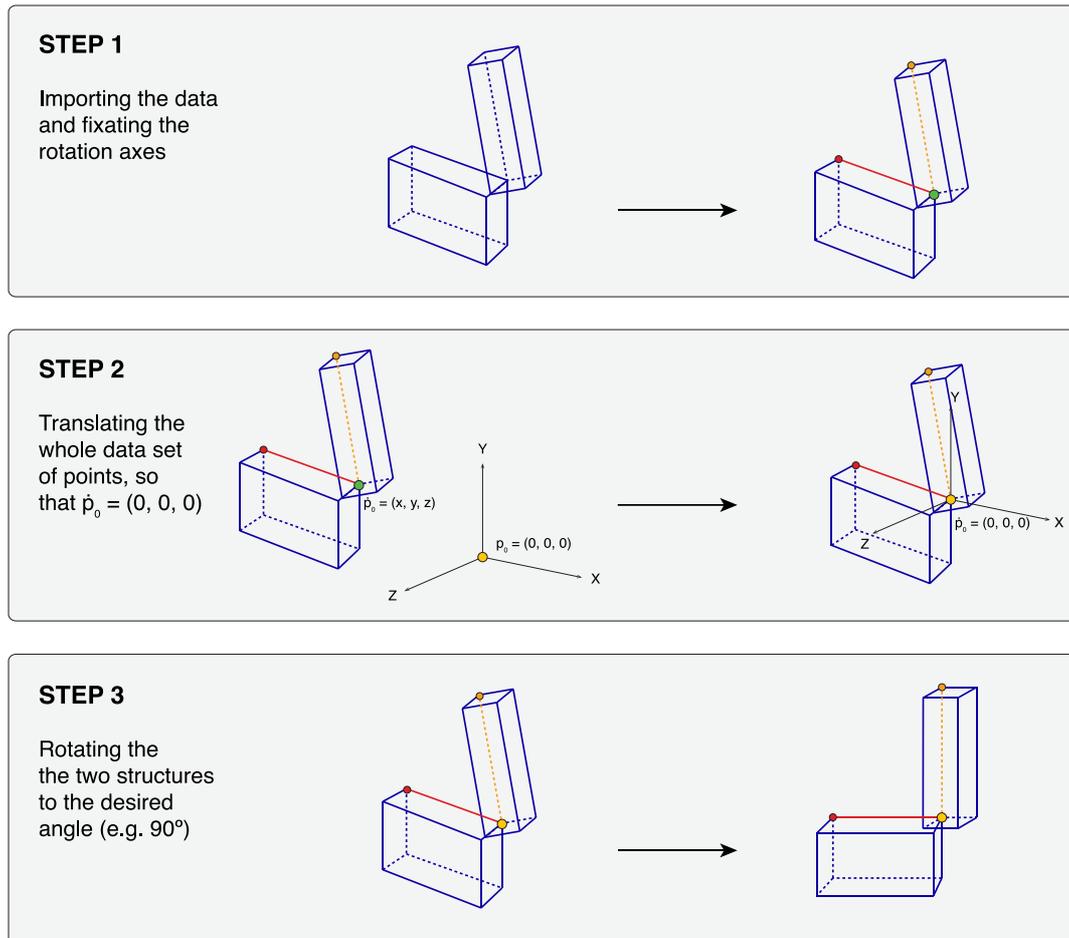
282

283 **FIGURES**



284

285 **Fig. 1.** (a) Application of the 3D rigid rotation method in three different scenarios, by rotating
286 articulated structures to a standardised position relative to each other. (b) Rotation method exemplified
287 by depicting the plane spanned by the already translated point p_M and A . Please note that p_0 depicts
288 the origin point $(0, 0, 0)$. The rotated resulting point r_M , vectors u and v , and angle θ are also depicted.



289

290 **Fig. 2.** Overview of the steps required in ShapeRotator, in order to rigidly rotate two articulated
291 subunits. Step 1: importing the data and fixating the rotation axes; Step 2: translating the whole data set
292 of coordinates or points so that the main selected point $\tilde{p}_0 = (0, 0, 0)$, with the function *translate()*; and
293 Step 3: rotating the two structures to the desired angle (e.g. 90 degrees), with the function *rotator()*.