

# Shape Trajectory Analysis Using Procrustes Analysis and VARMA Models



K. James Soda<sup>1</sup>, Dennis Slice<sup>1</sup>  
<sup>1</sup>Department of Scientific Computing  
 Florida State University



## Introduction

-The Procrustes paradigm of morphometrics provides a four step work flow for shape analysis: 1) Collect landmark data, 2) Align shapes via Generalized Procrustes Analysis, 3) Analyze resulting data via multivariate statistics, 4) Visualize data (Adams et al. 2013)

-A *shape trajectory* is a time-ordered set of shapes that an organism assumes during some behavior or process. Since after alignment, shape trajectories are functions, rather than points, in shape space, the third step of the paradigm cannot be implemented without bias.

-Shape trajectory data is similar to outline data, in that the unit of interest is a continuous function. To analyze outlines, morphometricians often approximate the outline using a combination of one or more basis functions and use the parameters of these functions as proxies for the outlines themselves. These parameters define a single point in parameter space, thus allowing for statistical analysis of the original outlines (Rohlf 1990).

-Vector Autoregressive-Moving Average (VARMA) models are a family of statistical models used to represent a vector-valued time series. Each model has the following generalized structure:

$$\Phi_0^\#(\mathbf{Y}_t - \mu) - \sum_{j=1}^p \Phi_j^\#(\mathbf{Y}_{t-j} - \mu) = \Phi_0^\# \epsilon_t - \sum_{j=1}^q \Theta_j^\# \epsilon_{t-j}$$

Put briefly, the left-hand side of a VARMA model expresses the value of the time series at time  $t$  ( $\mathbf{Y}_t - \mu$ ) as partially a weighted average of the previous  $p$  time steps. The remaining value of the series at time  $t$  is expressed on the right-hand side as a weighted average of a normally-distributed random vector ( $\epsilon_t$ ) and the random vectors that occurred over the last  $q$  time steps (Reinsel 1997).

-Here we fit VARMA models to simulated shape trajectories and then use the elements of the resulting coefficient matrices as a representation of the trajectories themselves. Using these representations, we attempt to differentiate between two qualitatively different shape trajectories.

## Methods

### Simulation Design

-Both simulations were written in Python ver. 2.7 and consisted of ten landmarks (see Fig. 1) (van Rossum and Drake 2001). The first time step in both trajectories began with the configuration seen in the first panel of Fig. 1A, and in both cases the simulated organism moved such that the most anterior and most posterior landmarks only moved along a single line. However, their lateral landmarks moved to either simulate peristaltic movement (Fig. 1A) or undulatory movement (Fig. 1B).

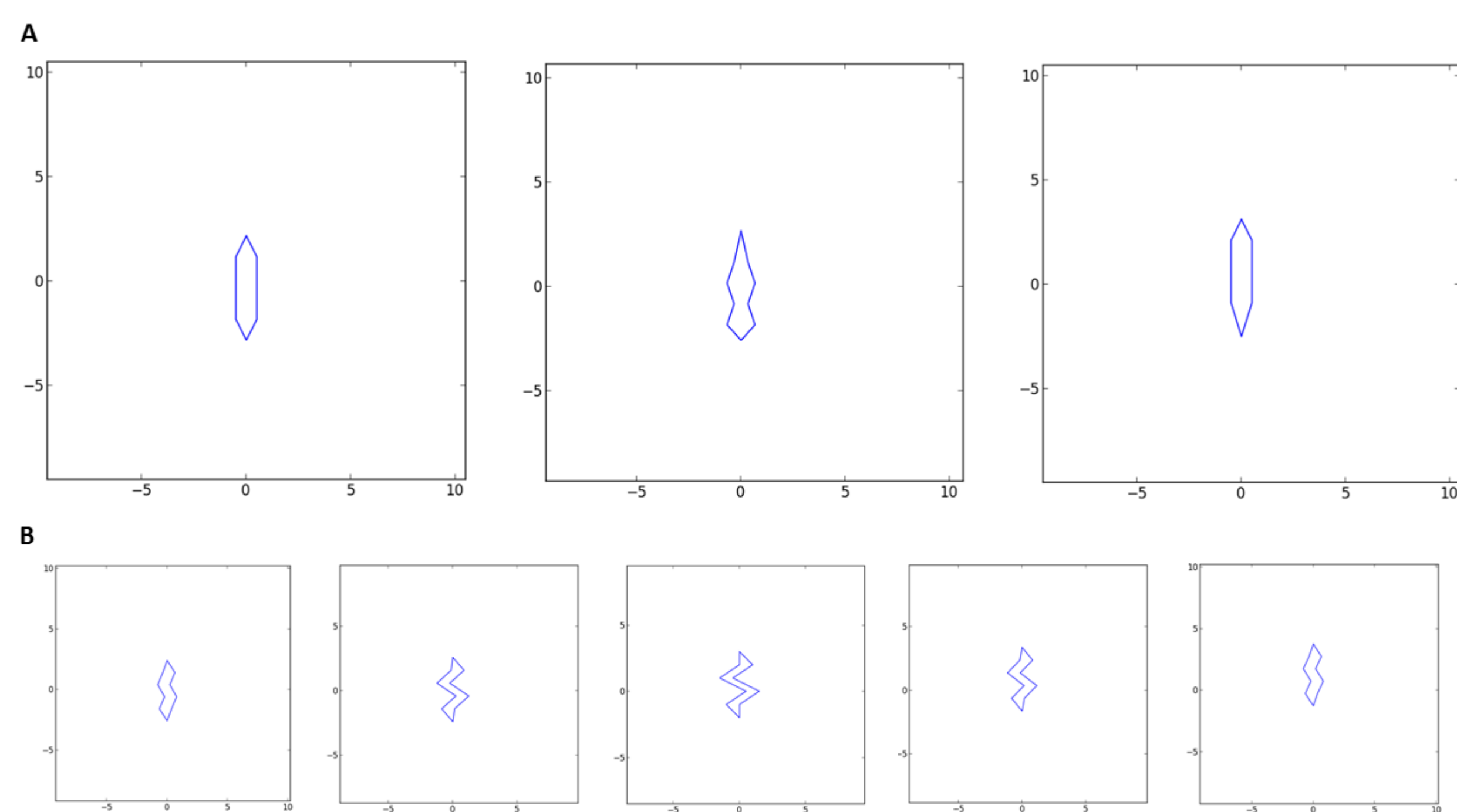


Figure 1. A) Example of three time steps from the peristaltic simulation, illustrating how the shape of the simulated organism changes as it moves. Notice that the lateral landmarks can only move along the x-axis until the most anterior landmark stops moving along the y-axis. B) Example of five time steps from the undulatory simulation, illustrating how the shape of the simulated organism changes as it moves. Notice that the lateral and most anterior landmarks can both move along the y-axis at the same time but that only the lateral landmarks can move along the x-axis.

## References

- Adams, DC, FJ Rohlf, and DE Slice. 2013. A field comes of age: geometric morphometrics in the 21st century. *Hystrix* 24: 7-14.
- R Core Team. 2013. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Reinsel, GC. 1997. Elements of multivariate time series analysis (2nd ed.). New York, NY: Springer.
- Rohlf, FJ. 1990. Fitting Curves to Outlines. In FJ Rohlf and FL Bookstein (Eds.), Proceedings of the Michigan Morphometrics Workshop. Ann Arbor, MI: The University of Michigan Museum of Zoology.
- SAS Institute Inc. 2002-2004. SAS 9.1.3 Help and Documentation. Cary, NC: SAS Institute Inc.
- Slice, DE. 2013. Morpheus et al (ver. 1.7) [software]. Available at <http://morphlab.sc.fsu.edu/software.html>.
- Sokal, RR and FJ Rohlf. 1995. Biometry (3rd ed.). New York, NY: W.H. Freeman and Company.
- Tsay, RS. 2014. MTS: All-purpose toolkit for analyzing multivariate time series (MTS) and estimating multivariate volatility models. R package version 0.32. <http://CRAN.R-project.org/package=MTS>
- van Rossum, G. and F. L. Drake (eds). 2001. Python reference manual. Virginia, USA: PythonLabs. Available at <http://www.python.org>.

## Methods (cont.)

### Processing and Model Estimation

-The shape trajectories were concatenated and aligned via Procrustes Analysis in Morpheus et al. ver 1.7 (Slice 2013)

-When describing shape trajectories, the size of the coefficient matrices in a VARMA model are dependent on the number of variables used to describe each shape in the trajectory. To reduce the number of variables that described each shape, and thus reduce the number of coefficients that require estimation, we applied principal component analysis to the data using R ver. 3.0 (R Team 2013). Then, with the help of routines in the R package MTS, we established the maximum values of  $p$  and  $q$ , the so-called orders of the models, that each trajectory could require when different numbers of principal components were used to describe each shape (Tsay 2014).

-With the first two principal components describing each shape, we built multiple VARMA models using the VARMAX procedure in SAS software ver. 9.3 (SAS Institute Inc. 2002-2004). We tried various model structures and compared the fit of each model using corrected Akaike Information Criterion. Based on a haphazard search of model space, we chose a VARMA(2,2) model to represent each trajectory.

-Based on estimates provided in SAS software, we compared each element in the model for the peristaltic simulation to its corresponding element in the model for the undulatory simulation using a Welch t-test in R.

## Results

Table 1. Results from Welch t-tests comparing corresponding elements of the coefficient matrices for the peristaltic and undulatory simulations. The first table corresponds to matrices weighting vectors from the previous time step, and the second table corresponds to matrices weighting vectors from two time steps before time  $t$ . VAR corresponds to matrices weighting previous shapes, and VMA corresponds to matrices weighting previous random vectors. Asterisks indicate significant results after a Bonferroni correction for each component of the model ( $t_{critical} = 2.958783$ ) (Sokal and Rohlf 1995).

Element of Matrix	VAR (Peristaltic/Undulatory)	t-value	VMA (Peristaltic/Undulatory)	t-value
<b>1</b>				
1,1	1.04763/1.66579	-6.704416*	0.42242/0.83115	-4.202830*
1,2	0.00008/0.66220	-1.756473	0.00018/0.66770	-1.618387
<b>2,1</b>	-570.73680/-0.00018	-15.859051*	-599.32678/-0.00001	-9.381632*
2,2	0.16073/0.81781	-5.847210*	-0.04857/0.64501	-6.138377*
<b>Element of Matrix 2</b>				
1,1	-0.13385/-0.97719	9.201428*	0.11125/-0.13428	4.8698095*
1,2	0.00023/-0.67809	2.633741	0.00023/-0.03932	0.3095648
2,1	431.98980/0.00001	17.011872*	146.03884/0.00007	58.7901185*
2,2	0.38605/-0.93085	22.537573*	0.53328/-0.31113	10.7108239*

Table 2. The highest possible orders for VARMA models that describe the undulatory (to the left) and peristaltic (to the right) simulations when varying numbers of principal components describe each shape in the trajectory.

PCs Retained	Highest Order Possible
10 PCs	1/1
5 PCs	1/1
4 PCs	1/1
3 PCs	4/2
2 PCs	5/2

## Conclusions

-There appears to be an inverse relationship between the number of principal components used to describe shapes in a trajectory and the number of coefficient matrices that could be required to adequately describe that trajectory. This in practice could lead to a trade-off between the size of coefficient matrices and the number of coefficient matrices.

-The coefficients in VARMA models allow for statistical differentiation between the undulatory and peristaltic simulations. At the moment, such comparisons are limited by the software available for building VARMA models. The development of our own routines could vastly improve the sophistication and power of statistical comparisons.