

# SPASOs v.1.0: User Guide

## 1-. Overview

PASOS (Phylogenetic Analysis of Shape OntogeniesS) is a method to infer ancestral shape ontogenies and changes in developmental timing along a phylogeny departing from shapes described as landmark configurations. The approach is rooted in previous developments for the analysis of landmark data in a phylogenetic context with parsimony as optimality criterion (Catalano et al., 2010, Goloboff & Catalano 2011; Catalano & Goloboff 2012). The ontogenetic trajectory is delimited by starting and end points common to all species under analysis (double stage standardization Alba, 2002). The method allows analyzing ontogenetic trajectories with shapes recorded at different moments of the trajectories with age defined as either *a priori* defined stages or continuous time. This method is implemented in SPASOS (Software for Phylogenetic Analysis of Shape OntogeniesS), a windows command line program written in C language and open source. The program reads the data from a *tps* file including all the specimens under analysis. The reference tree should be given in parenthetical notation in a different file. While the inference of changes in developmental timing is performed in SPASOS, the optimization of landmark data on the tree is performed in TNT (SPASOS launches TNT). The program produced a series of output files that allow different *a posteriori* processing and visualization of the results.

**NOTE:** it is imperative to read the paper describing the method (Catalano *et al.* 2018) before running the analyses.

## 2-. Input Data

**Landmarks:** The specimens of all the species should be included in the same *tps* file. The ID of each specimen should include the species name, optionally followed by a specimen number identifier, with a slash symbol ("/") separating the species and specimen identifier. The age of each specimen (time or *a priori* defined stages) is indicated in a line preceded by the string "AGE=". If landmarks are in three dimensions, the string indicating the number of landmarks should be "LM3=" instead of "LM=".

### Example

```
LM= 5
-0.1738154 0 -0.05491769
-0.1523607 0 0.01276399
-0.1282652 0 -0.03952946
-0.09372994 0 -0.02876725
-0.09850838 0 0.06603539
ID=Panthera_leo/456
AGE=5
```

When working with *a priori* defined stages, the youngest stage should be 0 and the rest of the stages should be numbered consecutively. Each species should be represented by at least one individual per stage.

**Tree:** the topology considered to map the landmark data is read from a text file with the tree in TNT parenthetical notation and preceded by the string "tread".

### Example

```
tread  
(one (two (three four ) ) ) ;
```

## 3-. *Running the analysis*

In order to run the analysis, executables of SPASOS and TNT should be in the same folder. TNT can be downloaded here: <http://www.lillo.org.ar/phylogeny/tnt/>.

### *Mandatory arguments:*

- l "myinputfile.tps" Name of *tps* input file. If the file is not in the same folder than SPASOS, give the full path.
- t "mytreefile.tre" Name of tree file. If the file is not in the same folder than SPASOS, give the full path.
- a *cont* / *pre* (N). Indicate if age is given as a continuous variable (*cont*) or as predefined stages (*pre*). If age is given as predefined stages, the number of stages (N) should be indicated after the string "pre". In the case of age given as continuous variable, by default SPASOS divides the ontogenetic trajectories in the maximum number of stages without missing data. Optionally, the user can indicate the number of stages after the string "*cont*".

### *Optional arguments:*

- l Generate a txt file listing specimens included in each stage for all the species.
- o Infer ancestral ontogenetic trajectories in the original frame of comparison (i.e. force no change in developmental timing in all branches).
- p N Define a penalty level (0 no penalty, 1 lowest/5 highest). Default 1
- r N Do N replicates of resampling.
- u "myprefix". Give a prefix for output files.
- v Do not generate an svg file with implied alignment of ontogenetic trajectories.
- x Assign a penalty equal for all changes in developmental timing irrespective of the number of stages implied in the change.
- y Calculate decay support values.

Example: perform an analysis with landmark data included in the file *cats.tps*, the tree in the file *cats.tre*, age given as continuous variable. Calculate decay values and perform resampling analysis using 100 replicates.

```
spasos -i cats.tps -t cats.tre -a cont -y -r 100
```

### ***Changes in developmental time inferred in SPASOS***

The changes inferred by PASOS depend on how the age is given. If age is given as a priori defined stages, only shifts can be inferred. If age is given as continuous variable, it is also possible to infer stretches of the trajectories. Except for shifts, calculating the score of each change in developmental timing requires re-sorting the individuals in a different number of stages. A= ancestor; D= descendant.

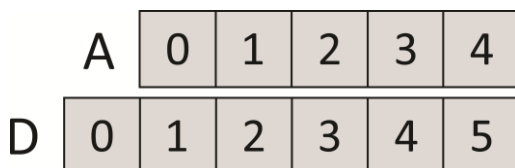
**Shift (Sh)**: displacement of the complete trajectory without changing its extension. A negative shift implies that younger stages of ancestor are matched with older stages of the descendant. A positive shift implies that older stages of ancestor are matched with younger stages of the descendant.

*Example: Sh+2*



**Stretch End (StrS)**: Extension or retraction at the start of the descendant trajectory while the matching at the offset is not modified. A positive stretch implies that younger stages of the descendant are matched with older stages of the ancestor at the start of the trajectory.

*Example: StrS-1*



**Stretch End (StrE)**: Extension or retraction at the end of the descendant trajectory while the match at the onset is not modified. A positive stretch implies that younger stages of the descendant are matched with older stages of the ancestor at the end of the trajectory.

*Example: StrE+1*

A	0	1	2	3	4	
D	0	1	2	3	4	5

**Double stretch (StrD):** Extension or retraction at both limits of the descendant trajectory. At both ends a positive value implies that older stages of ancestor are matched with younger stages of the descendant.

*Example: StrD+1+2*

A	0	1	2	3	4		
D	0	1	2	3	4	5	6

### ***Penalty***

SPASOS assigns a penalty cost for the changes in developmental timing that is proportional to the extent of the modification. The user can choose different penalty levels, from 0 (no penalty) to 5, being 1 the default value. In addition, it is possible to assign a penalty cost that is independent to the extent of the modification (argument – x). Penalty calculation is described in Catalano *et al.* (201X).

### ***Support values***

#### ***Resampling***

SPASOS calculates a measure of support of the changes in developmental timing inferred on the tree using a resampling procedure. The program repeats the analysis 100 times, each replicate assigning a probability of deletion of 1/4 to each specimen, taking into account that each stage should include at least one specimen. Resampling frequencies of each inference of change in developmental timing.

#### ***Decay index***

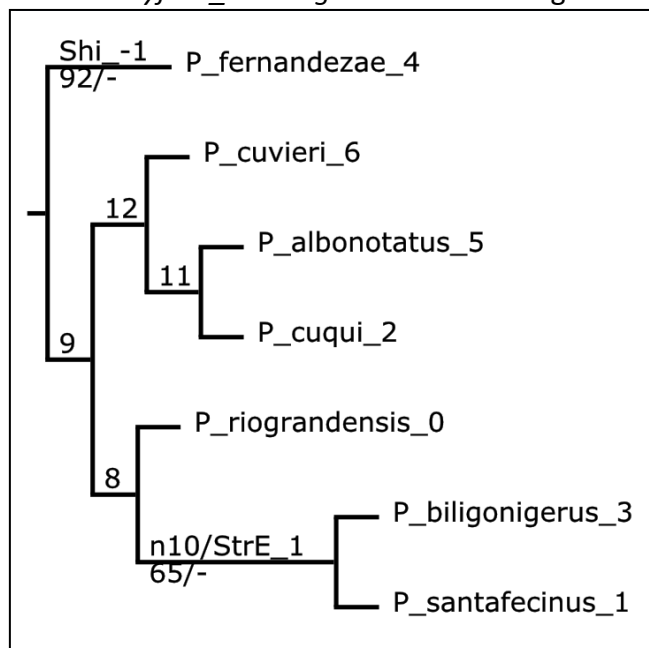
SPASOS can also calculate a decay index that determines how strong is the evidence that supports a certain inference of change in developmental timing. SPASOS reanalyses the data considering increasing penalty values (increase of 10% each time). Hence a value of 0.2 indicates that the transformation originally inferred is no longer retrieved when the analysis is repeated considering a penalty 20% higher.

## ***4-. Output files***

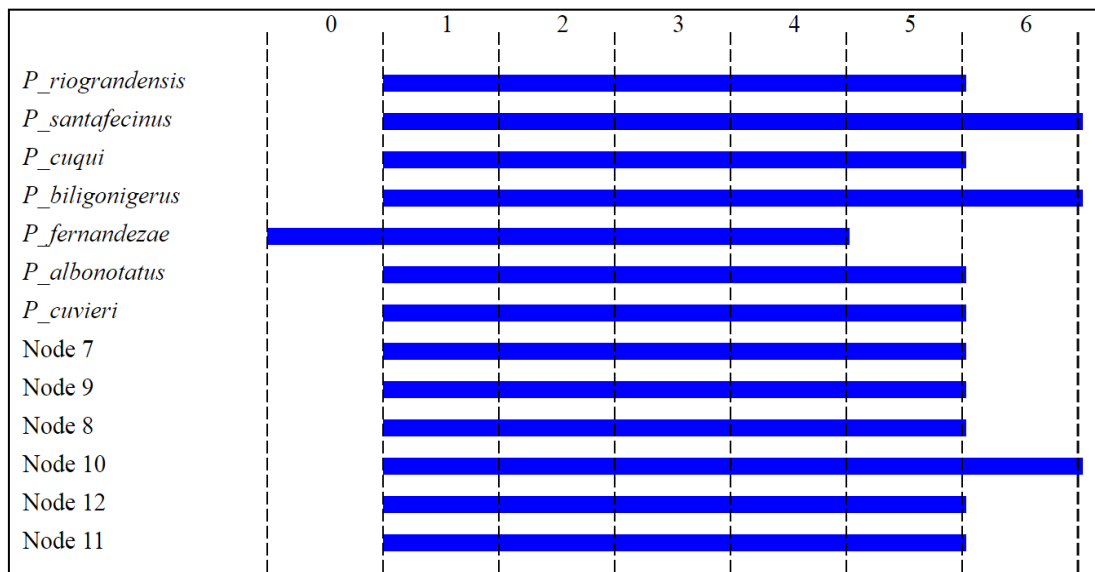
SPASOS produces a series of output files that allows visualization a post-processing of the results:

```
mxram 500;
xread
7 7
6[landmark 2d]
P_riograndensis_0 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? | -0.339543,0.038721 -0.309270,0.051003 -0.2531
P_santafecinus_1 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? | -0.325279,0.042995 -0.289385,0.056322 -0.23134
P_cuqui_2 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? | -0.332770,0.069862 -0.285255,0.067690 -0.241346,0.035
P_billigonigerus_3 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? | -0.346677,0.049991 -0.299577,0.058565 -0.2476
P_fernandezae_4 -0.352735,0.047693 -0.315450,0.057698 -0.258433,0.051900 -0.223462,0.096554 -0.004082,0.0
P_albonotatus_5 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? | -0.314130,0.052548 -0.273547,0.068071 -0.230585
P_cuvieri_6 ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? | -0.295088,0.042667 -0.269477,0.042727 -0.235721,0.0
;
```

4.3. A svg file including an image of the labelled tree described in 4.2. The file is named “myfile”\_tree.svg. Node numbering follows that in TNT files



4.4. A *svg* file showing a representation of the implied alignment of the ontogenetic trajectories derived from PASOS. Bars show the span of each trajectory. The file is named “*myfile*”\_traj.tre. Node numbering follows that shown in the *svg* tree file.



4.5. A *tps* format file that includes the shape of each stage for each ontogenetic trajectory (terminal and ancestral). Useful for post-processing.

```
LM=5
-0.339543 0.038721
-0.309270 0.051003
-0.253174 0.048396
-0.233481 0.107534
-0.034224 0.069229
ID=P_riograndensis_Stg_1
```

4.6. (Optional). A *txt* file listing specimens included in each stage for each species. The file is named “*myfile*”\_specimens.out

```
Species,Specimen,Stage
P_riograndensis,0,0
P_riograndensis,4,2
P_riograndensis,5,3
P_riograndensis,6,2
P_santafecinus,39,0
P_santafecinus,40,0
P_cuqui,54,0
P_cuqui,55,3
P_cuqui,56,2
P_biligonigerus,112,1
P_biligonigerus,113,2
P_biligonigerus,114,1
P_biligonigerus,115,2
```

## *5-. References*

Alba D.M. 2002. Shape and stage in heterochronic models. In: Minugh-Purvis N., McNamara K.J., editors. Human evolution through developmental change. Baltimore: The Johns Hopkins University Press. pp 28-50.

Catalano S.A., Goloboff P., Giannini N. 2010. Phylogenetic morphometrics (I): the use of landmark data in a phylogenetic framework. *Cladistics*. 26:539-549.

Catalano S.A., Goloboff P. 2012. Simultaneously mapping and superimposing landmark configurations with parsimony as optimality criterion. *Syst. Biol.* 61:392-400.

Goloboff P.A., Catalano S.A. 2011. Phylogenetic morphometrics (II): algorithms for landmark optimization. *Cladistics*. 27:42–51.