Mint: Modularity and Integration analysis tool for morphometric data. Version 1.6 (compiled 04/24/13)

Eladio Márquez
Mammals Division
University of Michigan Museum of Zoology

http://www-personal.umich.edu/~emarquez/morph/

Current address:
Department of Biological Science
Florida State University

Citations:

Márquez, E. J. 2008. A statistical framework for testing modularity in multidimensional data. *Evolution* 62:2688-2708.

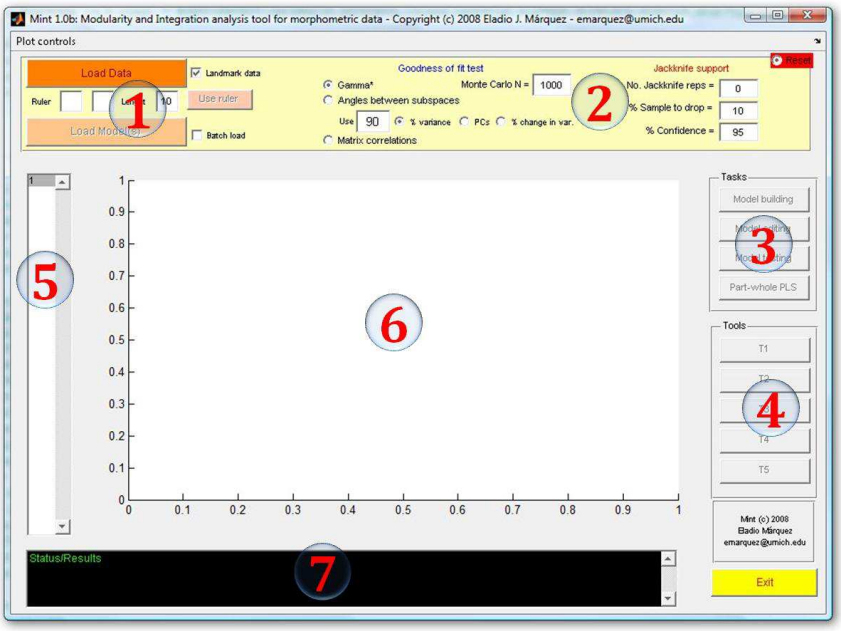
Parsons, K.J., Márquez, E.J., Albertson, R.C. 2012. Constraint and opportunity: the genetic basis and evolution of modularity in the cichlid mandible. *The American Naturalist* 179:64-78.

Reference Guide

Mint (Modularity and Integration tool for morphometric data) implements most of the procedures described in Márquez (2008) to test a priori models of variational modularity in multidimensional (mainly morphometric) data. The program accepts both 2-D landmark and non-landmark data, though landmark data are needed to get the most out of the visualization options in Mint. In addition to implementing goodness of fit tests for *a priori* models of modularity, Mint allows editing models and re-combining their modules to form additional, mixed models, thus facilitating the implementation of exploratory analyses of modularity and integration based on a heuristic model search.

The main areas of Mint's interface (see picture below) are (1) an input panel, (2) a parameter setting panel, (3) a **TOOLS** panel that defines Mint's main functions (i.e., build models, edit existing models, goodness of fit tests of models, and Partial Least Squares analysis of parts vs. whole structures), (4) a **TASKS** panel which allows access to functions specific to each "tool", (5) a *list element*, which is populated by models, modules, or landmarks/variables depending on the context, (6) a plot area, and (7) a text box where Mint prints tutorial hints, status reports, and result summaries.

The overall structure of Mint consists of three basic phases: first, landmark or non-landmark data are loaded, which resets all parameters and internal structures to their initial values; second, a model or set of models are loaded, built, edited, and/or combined; finally, these models are either tested against data, or used to regress individual modules on the entire dataset using PLS. Outputs from such analyses are presented as plot and textual descriptions, and result matrices can be saved as text files.



In the following guide, one of the datasets used in Márquez (2008) is analyzed in detail to show the functionality of Mint. It corresponds to a sample of 39 mandibles of the sigmodontine species *Sigmodontomys alfari*, on which 18 landmarks and 51 semi-landmarks have been digitized and superimposed. Prior to these analyses, data have been standardized to remove allometric variation (for details, see Márquez, 2008). All of the files and datasets mentioned in this guide have been included with this release of Mint, unless otherwise indicated.

Loading data

Before loading a dataset, the 'Landmark data' checkbox should be marked accordingly to ensure Mint processes the data correctly. Format for non-landmark data consists simply of a rectangular matrix with specimens or individuals as rows and variables as columns. For 2-D landmark data, Mint accepts a matrix ("XY") format, with consecutive pairs of columns corresponding to <*x*,*y*> coordinate pairs and rows correspond to individuals, an extended matrix format ("XYCS") in which a centroid size column is added as the last column of the set (included dataset uses this format), and the TPS format used by tpsDig and other programs in the tps series by F. J. Rohlf. Right now, Mint is able to recognize only simple forms of TPS files, in which the only present tags are "LM" and "SCALE". Presence of other tags may cause Mint to produce an error message and fail to load a dataset.

If two ruler points are included in data files for scale reference, their corresponding landmark numbers should be entered in the "Ruler" boxes in the input panel. Pressing "Use ruler" instructs Mint to scale all specimens so that they all have the same ruler length, after which the ruler points are dropped from further analyses.

After successfully loading a dataset, and prior to loading models, the only "tool" enabled in Mint is **MODEL BUILDING**, which is discussed below. If data consist of landmark coordinates, the plot area will show a GLS (Procrustes) superimposition on these data.

After loading the dataset analyzed for this guide (`jaw.dat`), the plot area will show the following graph:

Loading, building, editing, and combining models

Models are loaded in Mint as protocol files, which can be loaded individually or using a batch file. When the **BATCH LOAD** checkbox is marked, Mint expects a text file with has as many lines as models to be loaded, each line containing either the full directory path where the corresponding file is located, or just the file name if this is located in the same folder where Mint is currently running. An example batch file has been included in this release (`jaws.model.batch12.txt`) which instructs Mint to load the protocol files `jaws.model.XX.txt`, also included.

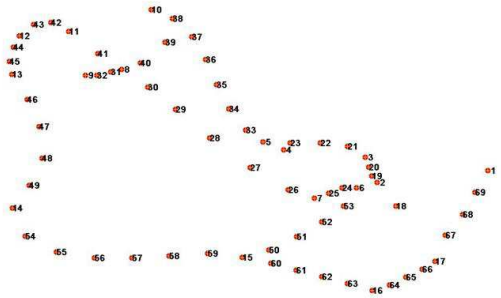
Each model protocol file must contain two columns: the first one must contain numeric labels for each "module" or "part" being tested, and the second must match these labels to individual variables or landmarks. To be recognized as valid, (1) each protocol file must list all landmarks/variables (in the second column), and (2) both landmark/variables and modules/parts must be numbered sequentially starting at 1. In Mint, ***Modules can overlap***, which means that variables/landmarks can simultaneously belong to two or more modules. In its present version, however, **Mint cannot handle nested modules**, i.e. those in which all of the variables of a module overlap with some of the variables of another module (see below for details).

Modules comprised of landmark should contain at least 3 landmarks to qualify as shapes, although ignoring this requirement will not prevent Mint from testing a model's goodness of fit, and this may be sometime desirable to group landmarks whose module association cannot be determined from data (e.g., there is no variance associated to them). However, part-whole PLS analyses involving a module/part with less than three landmarks will not run.

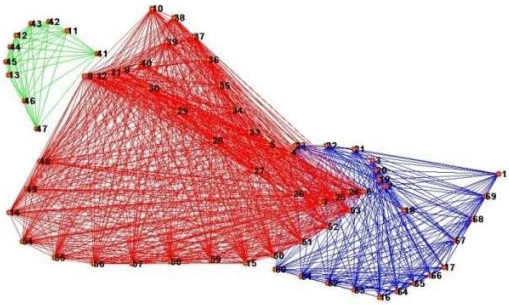
Twelve models have been included with this release of Mint (files `jaws.model.XX.txt`), corresponding to the 12 alternative hypotheses derived from developmental theory in Márquez (2008).

Standardization of the statistics used in the approach used by Mint requires the inclusion of a "null" model representing total absence of integration (where each variable comprises its own "module"). This model produces a diagonal-only covariance matrix where all covariances among variables/landmarks equal zero. It is not necessary to include a null model among the loaded alternatives, but if it is, **it must be loaded only once, as the first model in the list**, in order to be properly recognized by Mint. If, upon loading, Mint does not detect the first of the loaded protocols to correspond to a diagonal-only model, *it will create one automatically* and will assign the first position to it.

Loading the 12 models included with this release (using the batch file for added convenience) will result in Mint populating the list element with 13 models. When using landmark data, selecting models from the list will instruct Mint to show a representation of such models in the plot area. In these representations, landmarks that belong to the same module are mutually joined by colored lines, where different modules are distinctly colored. For example, loading the included 12 models and selecting the first one in the list (i.e., the "null" model, automatically added by Mint) will produce the following plot:



in which there are no modules. Likewise, selecting model #9 will produce this plot:



which depicts three modules using different colors. Visualizations like these are accessible at all times while none of the **TOOLS** buttons are selected.

Further model loads will not erase previous ones, but will add them to the list of currently loaded models, in the same order as they are being loaded.

Models can also be built from scratch or edited using the appropriate functions from the **TOOLS** panel. Thus, selecting **MODEL BUILDING** creates a "blank model" where no modules are

defined, and populates the list element with variables or landmarks. The **TASKS** panel buttons are then modified to show the functions: **SELECT MODULE**, **UNDO LAST MODULE**, **DELETE NEW MODULES**, **APPEND MODEL**, and **SAVE MODEL**. The building process works as follows.

First, on the list element, select groups of variables or landmarks using the Ctrl and Shift keys as usually done for multiple selections in Windows. Alternatively, when using landmark data, clicking on the plot area will change the cursor so that a polygon can be drawn around landmarks using left-clicks of the mouse. Right- or double-clicking after a polygon has been drawn will cause Mint to select the enclosed landmarks in the list. After a selection has been made, pressing **SELECT MODULE** will mark the set as a module in the model and will display it in the plot area.

After at least one module has been selected, clicking on **UNDO LAST MODULE** will erase the previously selected module, which can be done until there are no more modules selected. Clicking on **DELETE NEW MODULES** will erase all modules entered so far.

Finally, the functions **APPEND MODEL** and **SAVE MODEL** instruct Mint to include the model among the current list of models or to save it to a text file as a protocol file, respectively. Note that saving a model to a file does not automatically add the model to the current list, or vice versa. In order to append or save a model, each landmark/variable must belong to at least one module, or Mint will produce an error message.

The **MODEL EDITING** tool allows modifying one of the currently loaded models. Upon selecting this tool, the list element is populated with the currently loaded models, whereas the **TASKS** panel offers five new functions, namely, **SELECT MODEL**, **RESET SELECTION**, **REPLACE MODEL**, **APPEND MODEL**, and **SAVE MODEL**. At the beginning, only the first of these functions is enabled. Upon selecting a model from the list, the same button will turn to the function **SELECT MODULE** and, accordingly, the list will be populated with the modules of the selected model (indicated in the status text box at the bottom). From this point on, it is also possible to use the **RESET SELECTION** function to start over the editing process.

After selecting a module, the list will now be populated with all of the variables or landmarks in the dataset, and those corresponding to the selected module will be pre-selected. It is now possible to modify this selection in the list or (when using landmark data) by clicking and drawing directly on the plot as described above for model building. When finished, clicking on **SELECT LANDMARKS** will mark the selection as the new module, and will activate the other tasks associated to this tool. **REPLACE MODEL** will delete the model being currently edited and replace it with the edited version, which will be then assigned the same label/number for further analyses; **APPEND MODEL** will add it as the last model of the current list, with a new label/number, and **SAVE MODEL** will prompt the user to save the model as an individual protocol file.

Lastly, Mint offers the option to mix the individual *modules* defined within loaded *models* to form new models for analysis. This is useful under a number of scenarios, such as when hypothesizing patterns of modularity resulting from the cumulative effect of spatially localized developmental processes throughout ontogeny, or simply to increase the robustness of the

estimates of goodness of fit statistics (see below). The **COMBINE MODELS** task is accessible from the **MODEL TESTING** tool, and enabled after two or more models have been selected from the list element by clicking on the **SELECT MODEL** button. ***When searching for model combinations, Mint ignores model #1 (null model), as its inclusion prevents the search from converging***, as well as nested (hierarchical) modules (see justification below), but does an otherwise comprehensive search of all possible module combinations. Therefore, ***this search process can take quite long to finish***, depending on the number of models initially selected, during which regular status reports are printed in the *command window* that opens along with the main Mint interface. After the search is over, the newly found models are added to the list of currently loaded models and become immediately available for further analysis. It is highly recommendable to save loaded models at this point, by clicking on **SAVE OUTPUTS**, and then selecting **SAVE CURRENT MODELS**. Models are then saved as separate protocol files, to which Mint automatically appends a numeric label.

Testing goodness of fit of models

In goodness of fit (GoF) tests we are interested in assessing whether a pre-defined model or hypothesis is good enough to explain variation in a dataset. In practice, we want to test the null hypothesis that any discrepancy between data and model is small enough that it could be due to chance. In Mint, the interest is on testing the GoF of models depicting tight associations within integrated sets of traits—variational modules—and no associations with traits outside of those sets. The three essential components of a GoF test are:

1. An expectation for the data under the assumption that the is model is true;
2. A GoF statistic that measures the similarity between this expectation and the data;
3. A distribution for this statistic under the null hypothesis that expected and observed patterns differ only by chance.

Expectations from models

The rationale used by Mint to generate expectations for the data given the hypotheses under scrutiny is described in detail in Márquez (2008). Basically, the aim is to compute the covariance matrix that the data would be expected to produce if the model being tested was true and variances and covariances were measured without error. In models of modularity, the expectation is that phenotypic variables or morphological regions comprising a variational module are mutually integrated but statistically independent from other such modules. Because this expectation takes into account both within- and between-module associations, it seems reasonable to state them in terms of full covariance matrices, so that GoF statistics (see below) can assess both levels of integration simultaneously. The alternative, i.e. treating expected modules as isolated data partitions, has the disadvantage that tests must focus only on between-module associations, e.g. by asking whether they are small enough to justify defining the corresponding partitions as modules. To obtain the covariance matrices expected under models of modularity, Mint thus assumes that the data themselves have a modular structure, by partitioning the entire data space into orthogonal subspaces (i.e. modules) and computing a covariance matrix based on this modified data structure.

In practice, partitioning the data space into modular subspaces is carried out by stacking together as many replicates of the same dataset as modules there are in a model, so that there is a full dataset per module, and then making each element not belonging to a module equal to zero in its corresponding dataset copy. For example, if we wish to compute an expectation from a model with three modules containing the coordinates for landmarks/variables 1-8:

[1 2 3] [4 5 6] [7 8]

where the numbers in brackets represent the landmarks/variables included in each module, we first partition the full dataset with *n* observations and eight variables into three subspaces, by forming the extended data matrix

$$\mathbf{X_0} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{7} & \mathbf{8} \end{bmatrix}$$

in which each element corresponds either to one of the variables (vectors **1**-**8**) or to a vector of zeros, each of length *n* (Márquez, 2008).

In the general case, the covariance matrix $\mathbf{S_0}$ obtained from this matrix equals the observed covariance matrix in all elements but those corresponding to covariances between modules, which equal zero. When using landmark data, however, a GLS superimposition of $\mathbf{X_0}$ is carried out prior to computing its covariance matrix, to take into account the covariances normally induced by this step (Walker, 2000).

The process described above is valid for models in which modules do not overlap (i.e. do not share variables/landmarks), as they are expected to have covariances equal to zero. When two modules overlap, possibly due to two or more different processes affecting the same trait or region simultaneously or at different points during ontogeny, such modules cannot be longer expected to give rise to orthogonal subspaces. If modules overlap only partially, so that each still contains mutually uncorrelated elements, then their subspaces will be oriented obliquely in the full data space. In its current implementation, Mint simulates module (partial) overlap by distributing the variance of each shared element equally among the overlapping modules, so that their variances are unchanged in the full covariance matrix (see Márquez, 2008 for further details). Future versions will implement alternative ways for partitioning variation at overlapping regions.

Currently, Mint ignores expectations from modules that overlap completely (i.e., nested or hierarchical modules), because in such cases modules define subspaces embedded within larger subspaces, so that in the present approach they appear as integrating the same module. Testing whether a module is nested within another module may require using a different kind of approach, in which covariances are fine-tuned to reflect the differences in integration expected between embedded and embedding modules.

Goodness of fit statistics

In Mint, GoF tests are carried out by comparing observed and expected covariance matrices, for which three choices of GoF statistics are offered: γ* (=Gamma*, Richtsmeier et al. 2005), angles between the subspaces spanned by these covariance matrices (Zelditch et al. 2006), and matrix correlations (Dietz 1983). Only one of them (γ*), however, has been fully standardized to remove artifacts introduced by differences in the number of fixed parameters in models. As discussed in Márquez (2008), this standardization is facilitated by the fact that γ* scales linearly with the number of "zero elements" (inter-module associations) in covariance matrices, which allows using simple linear regression as the standardization technique. The implication is that **the robustness of γ* is largely a function of the number of models tested simultaneously**. If angles between subspaces or matrix correlations are used instead of γ*, additional observations should be carried out to make sure that results are not overly influenced by the number of fixed parameters in models (e.g., by plotting the statistic value against the number of zero elements in models).

In Mint, γ* standardized values are computing by (1) scaling and (2) regressing γ values on the number of zeros of each model, with

$$\gamma = \text{trace}\ \{(\mathbf{S} - \mathbf{S_0})(\mathbf{S} - \mathbf{S_0})^T\}$$

(Richtsmeier et al. 2005), where **S** and $\mathbf{S_0}$ are the observed and expected (i.e., derived from $\mathbf{X_0}$) covariance matrices, respectively. Scaled γ values are obtaining by dividing each model's γ by the maximum γ value, γ_{max}, which is obtained by comparing the data against the null (diagonal-only) model. Therefore, scaled γ values are bound to the interval [0, 1]. This standardization is done to allow comparing results from different samples (i.e., species), and does not change the outcome of the regression within a sample. Note that null distributions (see below) are computed based on unstandardized values.

Next, each scaled γ value is regressed on the number of zeros in each model, representing expected associations between variables postulated to belong to distinct modules, to remove the effect of the number of estimated parameters. The final standardized statistic is defined as the residual $\gamma^* = \gamma - f(z)$, where $f(z)$ represents the linear function relating the values of γ computed from all possible models of modularity to their corresponding counts of zero elements, z. Even though it would be computationally unfeasible for most studies to include all possible models, the fact that scaled γ values are restricted to the interval [0, 1], where 0 corresponds to the observed covariance matrix and 1 to the null model of no integration, implies that $f(z)$ must also vary within these limits, which are sufficient to define the linear function for any given set of variables. Given a large random sample of models, with γ values symmetrically distributed about their mean, $E(\gamma) = f(z)$ and thus $E(\gamma^*) = f(z)$. Consequently, situations where $\gamma^* < 0$ correspond to models that postulate no integration (i.e., average covariances are hypothesized to be zero) for covariances whose observed values are indeed low, and conversely cases where $\gamma^* > 0$ correspond to models that postulate no integration for covariances whose observed values are actually large. Therefore, best-fitting models are those with the lowest γ* value. This approach, used in Parsons et al. (2012) differs slightly from the one used in Márquez (2008),

where $f(z)$ was estimated via regression using only the models included in a study, as opposed to *all* possible models.

The angle θ between two subspaces is the minimum angle with which the subspace spanned by a sample must be rotated to maximally align it with the subspace spanned by the other sample (Zelditch et al. 2006). Mint uses the formulation described in Zelditch et al. (2006: Appendix A) to determine the value of θ between expected and observed covariance matrices (standardized by its maximum value as described above for γ). This implementation requires an a priori definition of the number of axes used to define the subspaces being compared, to ensure that bases of the same size (i.e., number of principal components—PCs) are rotated. Because dimensionality of expected and observed matrices need not coincide, additional criteria must be used to determine the extent of the variation of each matrix that is included when computing the θ statistic. Mint offers three options for these criteria: (1) choosing number of PCs based on the percentage of the variance of a sample that they account for, (2) directly entering number of PCs, and (3) choosing PCs sequentially until the difference in variance accounted for by two consecutive PCs is smaller than a pre-determined value. Following, there is a detailed description of these choices.

When choosing the number of PCs that account for X% of the variance as a criterion, Mint attempts to include as many PCs as needed so that *at least* X% of the variance of both matrices being compared is included. Because the number of PCs has to remain constant, this means that one of the matrices may be represented by more than X% of the variation. However, it also means that the percentage of variation included in tests is limited by the size of the smallest subspace, i.e. the one with fewer dimensions. Consequently, it is possible to have a test in which 100% of the variation in a sample is compared against less than X% of the variation in another. Similarly, when the number of PCs is chosen directly, the first X PCs are used for computing θ, which means that different proportions of variance may be used for the samples being compared. Finally, choosing number of PCs based on the change in the proportion of variance accounted for by consecutive PCs could be in many instances a less arbitrary criterion, as it assumes that informative dimensions in a sample will be those that capture the best defined axes, whereas poorly defined, indistinct axes will most likely correspond to error variation, commonly assumed to be isotropic (Anderson, 1963). Selecting this option has the same limitations as using percentage of variance, and thus this criterion is likely to be met asymmetrically in both samples being compared.

The third option offered by Mint for a GoF statistic computes matrix correlations between expected and observed covariance matrices (r_M). As implemented herein, significance testing of matrix correlations is not based on Mantel tests, and thus this procedure does not permute covariance matrices directly. Significance of r_M values is tested using the same Monte Carlo procedure used for the other two statistics, and described below. In Mint, matrix correlations are carried out *excluding* the diagonal of the covariance matrices, and thus the minimum correlation (between data and the null, diagonal-only model) is always zero. Also, it is worth remembering that magnitudes of r_M values are interpreted so that higher values (closer to 1)

indicate similarity between data and model, whereas in the other two statistics (γ^* and θ), smaller values (closer to 0) are the ones that indicate similarity.

While all of these statistics are used for a common purpose, it is important to keep in mind that they do not measure the same properties of observed and expected covariation patterns. In fact, they measure quite different properties. Consequently, it is possible to obtain different and even contradictory results from their implementation, thus requiring further inquiry to determine the origin of such differences.

Test distributions

Whereas GoF statistics measure similarities between observed and expected covariance matrices, the null hypothesis that the value of this statistic is no larger than expected by chance is tested in Mint by comparing it against a null distribution for this statistic, i.e., a range of values for the statistic that can be obtained should the null hypothesis be true. Such test is rejected ($P < 0.05$) for sufficiently large differences between observed and expected covariance matrices (i.e., for large γ^* or θ values, or low absolute r_M values).

To compute a null distribution, Mint implements a parametric (Monte Carlo) approach, described in detail in Márquez (2008), in which N pseudo-random covariance matrices are generated from a Wishart distribution parameterized with the *expected* covariance matrix and the size of the sample being tested. A Wishart distribution is the probability distribution of the covariance matrix (actually, sums of squares and cross-products matrix) of a multivariate normal population, and in the present test simulates the range of possible matrices that could be obtained from the sampled populations should the model of modularity be true. To obtain a distribution of the GoF statistic of choice, Mint computes the value of the statistic between each random matrix and the expected matrix ($\mathbf{S_0}$). A P-value for the null hypothesis is then calculated by dividing the number of instances in which the GoF statistic computed between a matrix from the null distribution and the expected matrix is larger, in the case of γ^* and θ, or smaller, in the case of r_M, than the value for the statistic obtained between expected and observed matrices, by the total number of Monte Carlo replicates. Thus, significant P-values (< 0.05) correspond to cases in which expected and observed matrices are too different to be considered as part of the same distribution.

Jackknife support and model space

Before demonstrating this approach on the *S. alfari* data, an additional statistic, "jackknife support", is introduced. As shown below and discussed in Márquez (2008) alternative models of modularity are often similar enough that their discernment by Monte Carlo tests may require extraordinarily large sample sizes. Therefore, under many circumstances it may be preferable to focus comparisons among alternative models not just on which ones are significant, but also which ones provide a better fit to the data. For this purpose, Mint ranks the models from best to worse fit after running a test of several alternative models. In order to estimate the confidence of such ranks, Mint allows randomly resampling the original dataset to determine the proportion of random replicates in which model fit rankings are the same as those observed using the original dataset. These proportions, jointly termed jackknife support because resampling is

carried out by dropping a random set of individuals from the sample, provide thus a measure of confidence for all observed ranks, although interest will often be focused on the best fitting models.

Mint is instructed to compute jackknife support by requesting re-sampling of one or more replicates in the box labeled **NO. JACKKNIFE REPS**, and by specifying how many individuals to randomly drop in each replicate in the box labeled **% SAMPLE TO DROP**. In addition to computing jackknife support, Mint will calculate a confidence interval for the GoF statistic, using the level specified in the box labeled **% CONFIDENCE**.

The rationale described above for comparing alternative models based on their fit rankings can be further extended to interpret the full set of values of GoF statistics for a single dataset as a set of distances between the data and each model. The resulting vector of data-model distances can in turn be interpreted as a set of coordinates for the dataset within an abstract space defined in terms of model similarity, i.e., a "model space". GoF statistics defining this model space are included among Mint's standard outputs, and can be imported into external software for further analysis. In Márquez (2008), for instance, multiple species were compared using angles between those data-model distance vectors to assess interspecific similarity with respect to underlying patterns of integration and modularity, and resulting patterns were further compared against patterns of phylogenetic relatedness.

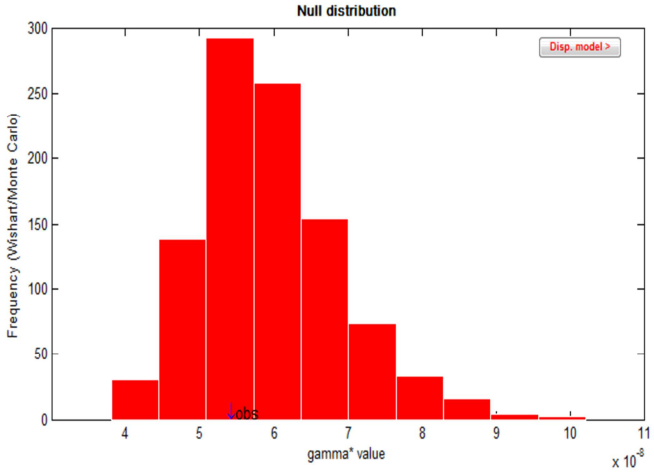
A worked example

After loading the data and model files included with this release (`jaw.dat` and `jaws.model.batch12.txt`, respectively), Mint should list a total of 13 models, with the null hypothesis of absence of modularity corresponding to model #1. To get the full set of models produced by searching all possible module combinations used in Márquez (2008), models #2 through #13 can be selected and submitted to such search by using the **COMBINE MODELS** tool accessible from the **MODEL TESTING** tool (see above for details). In this case, there are 1307 model combinations.

Upon selecting the **MODEL TESTING** tool, two or more models must be selected in the list element and marked for analyses by clicking on **SELECT MODELS**. For this example, analyses will be based on the full set of 13 models. Clicking on **RUN TEST** will instruct Mint to use the selected GoF statistic and Monte Carlo *N* to test all of the selected models against the loaded dataset. All of the runs shown here are based on $N = 1000$ Monte Carlo replicates. In most Mint functions, it is possible to check the command (black) screen that opens along with the main interface window for progress updates of current analysis and error messages.

Run #1: models tested using γ^*

After running the analysis, the list element of the interface shows the 13 models sorted from smallest to largest γ^* value. Best fitting model among the alternatives is thus #4, followed by #6, #7, #12, #9, #5, and so on. Worst fitting model is #1 (null model). The graph shown in the plot area, corresponding to the best-fitting model (#4):



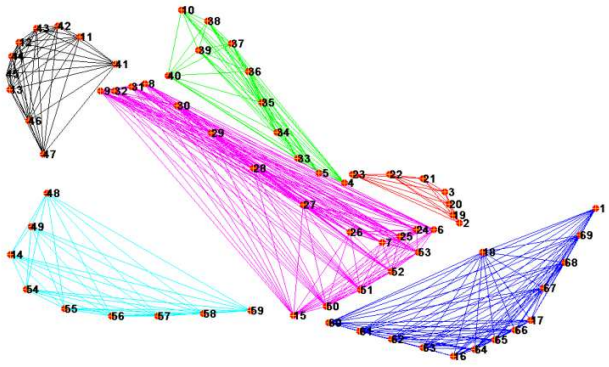
is a histogram of γ* values over 1000 (*N*) runs for the model selected on the list. The arrow on the *x*-axis indicates the approximate location of the observed value for the GoF statistic. The text on the bottom panel of the interface will provide a quick summary of results, in this case reading:

Currently selected model is #4, whose standard gamma value with respect to observed data equals -0.2952. P-value for the null hypothesis that the data are no more different from this model than expected by chance is 0.721 based on a Wishart/Monte Carlo test with 1000 replicates. The arrow in histogram (above) indicates the position of this model with respect to the null distribution for this hypothesis.*

The observed γ* value (-0.2952) is the lowest among the given alternatives, and a 95% confidence interval can be obtained by using the jackknife support task (see below). The value of γ* differs between the text and the histogram because the former has been standardized to facilitate comparisons across samples. The *P*-value shown for this model (=0.721) indicates that the observed γ* value between data and model is smaller than 72.1% of the γ* values simulated under the null hypothesis.

I some occasions, the P-value for a model may equal 1 (i.e., 100% of the simulated values exceed the observed γ* value). Although this seems counterintuitive, it should be remembered that this procedure does not yield a symmetric distribution and therefore the normal-like appearance of these results is most likely consequence of the relatively small sample sizes used herein. With large enough sample sizes (i.e., above 200), the distribution should take its proper form—that of a right-tailed distribution.

The button on the right top corner of the graph allows swapping between these results and the graphic representation of the selected model. In this case, model #4 is represented as follows:



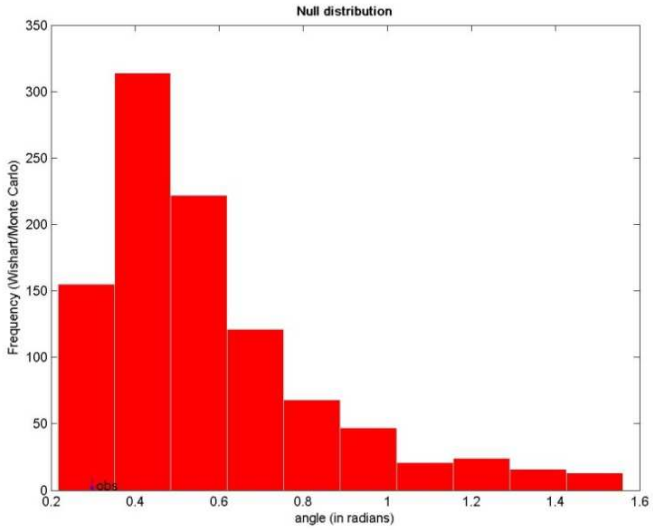
Selecting a different model will show a different set of results. For example, selecting the model ranked second (model #6), displays the following text:

Currently selected model is #9, whose standard gamma value with respect to observed data equals -0.26724. P-value for the null hypothesis that the data are no more different from this model than expected by chance is0.921 based on a Wishart/Monte Carlo test with 1000 replicates. The arrow in histogram (above) indicates the position of this model with respect to the null distribution for this hypothesis.*

In this example, none of the models, except for the null model is rejected (*P*-value > 0.05) when tested using the Monte Carlo approach.

Run #2: models tested using angles between subspaces

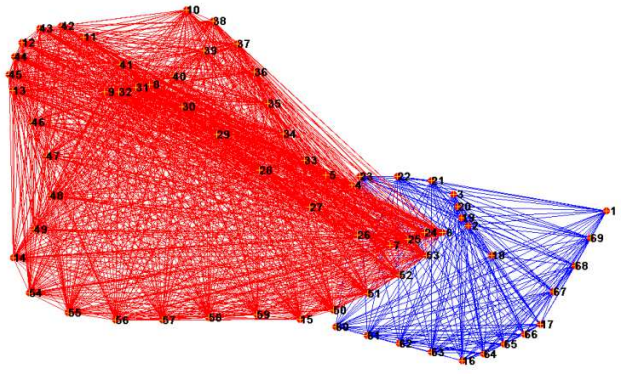
Running the analysis using the default parameters to compute angles between subspaces (No. of PCs is chosen so that at least 90% of the variance is included in comparisons) supports model #12 as the best fitting alternative (*P* = 0.96), followed by models #5, #10, #9, #11, and so on, with model #1 as the worst one (*P* = 0.019). The histogram for the angle statistic for model #12 looks like this:



in which angle values are given in radians. Similar to previous results, output text reads:

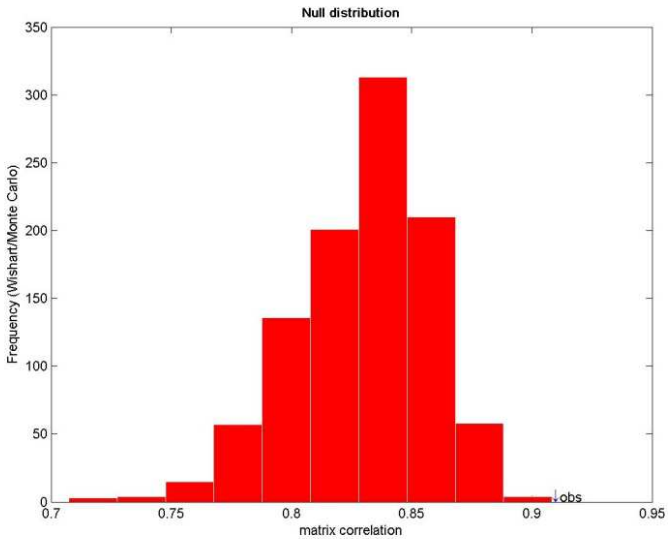
Currently selected model is #12, whose standard angle (in radians) with respect to observed data equals 0.19706. P-value for the null hypothesis that the data are no more different from this model than expected by chance is 0.961 based on a Wishart/Monte Carlo test with 1000 replicates. The arrow in histogram (above) indicates the position of this model with respect to the null distribution for this hypothesis.

Graphically, model #12 is represented as follows:



Run #3: models tested using matrix correlations

Analyses based on matrix correlations also return model #12 and #1 as best and worst fitting, respectively, while differing from previous tests regarding the rankings of other models. Histogram for model #12 is:



and the output text reads:

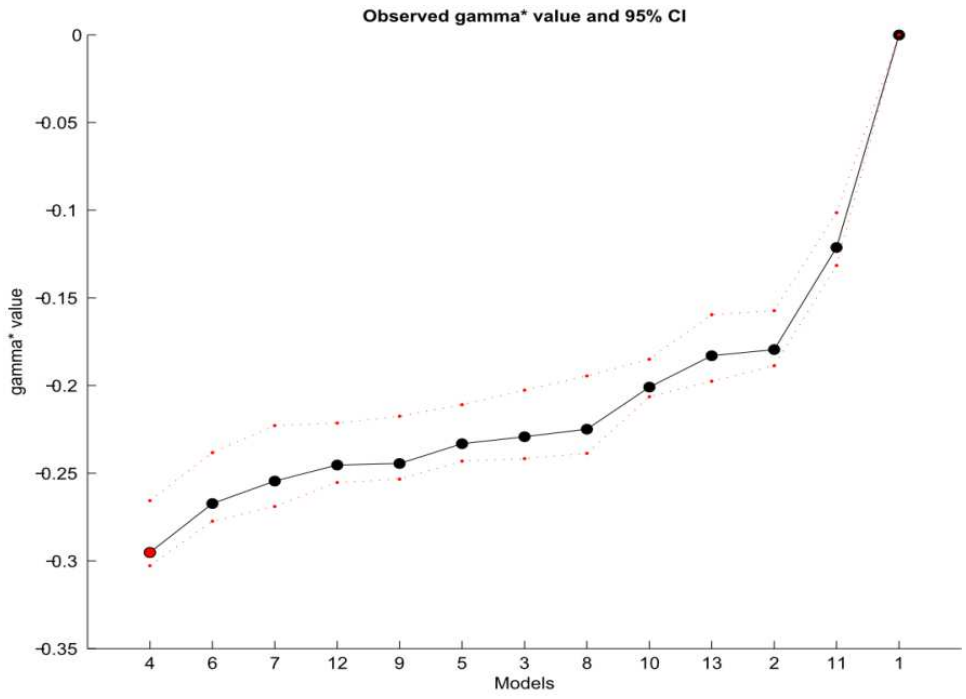
Currently selected model is #12, whose standard matrix correlation with respect to observed data equals 0.88548. P-value for the null hypothesis that the data are no more different from this model than expected by chance is 1 based on a Wishart/Monte Carlo test with 1000 replicates. The arrow in histogram (above) indicates the position of this model with respect to the null distribution for this hypothesis.

It seems worth noting that models #4 and #12 supported by these three methods are not mutually exclusive, as all of the putative modules of the former are completely embedded in one of the two modules of the latter. One interpretation of this discrepancy could be that methods based in observed angles and matrix correlations are more sensitive than the Monte Carlo approach to non-zero, but low, covariances among the six modules in model #4.

Jackknife support

To activate the **JACKKNIFE SUPPORT** task, the number of replicates entered in the box labeled **NO. JACKKNIFE REPS** should be at least one. For this example, 1000 replicates were run using γ* as the GoF statistic, dropping 10% of the specimens per jackknife replicate, and computing 95% confidence intervals for the statistic.

After analyses are finished, a plot of the GoF statistic and its confidence interval is shown for all models simultaneously:



As in previous analyses, a button is supplied with this plot for model visualization when using landmark data. The text output for the model occupying the first position is:

Currently selected model is #4, whose standard gamma value with respect to observed data equals -0.2952 (jackknife 95% CI = [-0.30273, -0.26556], based on 1000 replicates). Among the models included in the analysis, model #4 ranks #1. This rank was observed for this model in 100% of the jackknife replicates (i.e. jackknife support = 1).*

The reported jackknife support for this model (= 1) indicates that in each of the 1000 jackknife replicates for which a γ* value was calculated, model #4 was the best fitting alternative. In fact,

visual inspection of the plot shown above suggests that the confidence intervals of most models overlap, indicating that similar degrees of support are obtained for these models, suggesting that further refinement, e.g., via model combinations, is desirable to obtain better resolved patterns. For these models, Jackknife support estimates are:

Model #	Original rank	Jackknife support for orig. rank
4	1	1.000
6	2	0.876
7	3	0.457
12	4	0.260
9	5	0.409
5	6	0.536
3	7	0.522
8	8	0.649
10	9	0.925
13	10	0.670
2	11	0.700
11	12	1.000
1	13	1.000

These results indicate, for example, that model #8 is the *eighth best fitting* model in 64.9% of the jackknife replicates. In general, stronger support in more models and ranks indicates a more robust identification of the position of the dataset with respect to alternative models (i.e., in model space). Full set of outputs include, in addition to jackknife support for each original rank, the percentage of replicates in which each model occupies each rank (see below for details).

Saving results

After successfully running any of the available tests, Mint allows saving results by clicking on the **SAVE OUTPUTS** button. What saving options appear in the list that pops up depends on what analyses have been performed. Thus, running Monte Carlo tests adds two saving options (**SAVE TEST STATISTICS AND P-VALUES**, and **SAVE X% CONFIDENCE LIMITS OF NULL DISTRIBUTION**), jackknife support estimation adds four saving options (**SAVE JACKKNIFE SUPPORT VALUES**, **SAVE JACKKNIFE SUPPORT VALUES FOR ALL RANKS**, **SAVE MODEL RANKS FOR ALL JACKKNIFE REPS**, and **SAVE CONFIDENCE INTERVALS FOR TEST STATISTIC**), and searching for model combinations adds one option (**SAVE CURRENT MODELS**).

- **SAVE TEST STATISTICS AND P-VALUES** produces a matrix with as many rows as models were selected for tests and three or four columns. Models as sorted according to order in which they were loaded in Mint (or the order they appear in the batch file). Columns #1

and #2 contain values for the GoF statistic in its raw form and after standardization by its maximum possible value, respectively. The last column contains the *P*-value obtained as described above from Monte Carlo tests. Only when analyses are based on the γ* GoF statistic this output file contains four columns, the third one corresponding to γ* values (i.e., standardized γ corrected by the number of fixed parameters in models).

- **SAVE X% CONFIDENCE LIMITS OF NULL DISTRIBUTION**, where X equals the percentage of confidence specified for jackknife support tests, produces a matrix with three columns and as many rows as models were selected for analysis (sorted in the same order as they were input). The first column contains the percent confidence for the interval, whereas the second and third columns contain the lower and upper confidence limits for the raw GoF statistic, computed as the corresponding percentiles from the distribution of values obtained from Monte Carlo replicates.
- **SAVE JACKKNIFE SUPPORT VALUES** produces a data vector containing the estimates of jackknife support for the models selected for analysis (sorted in the sequence they were loaded).
- **SAVE JACKKNIFE SUPPORT VALUES FOR ALL RANKS** produces an *m* x *m* matrix (*m* = No. of models selected for analysis) where rows represent models and columns ranks, such that each cell contains the frequency with which each model was observed to occupy each rank in jackknife replicates.
- **SAVE MODEL RANKS FOR ALL JACKKNIFE REPS** produces a matrix in which columns contain the sequence of models sorted by the value of their GoF statistic in each jackknife run (i.e., each column contains the results associated to a single jackknife replicate).
- **SAVE CONFIDENCE INTERVALS FOR TEST STATISTIC** simply contains two columns with the lower and upper confidence intervals, as estimated from the jackknife replicates, for the GoF (standardized) statistic. These are the values sent to the plot after the resampling procedure is finished.
- **SAVE CURRENT MODELS** exports properly formatted individual protocol files for each model currently loaded. After selecting this option, a dialog prompts the user to assign a file name and folder location for the models being saved. Mint then appends a numeric label to the chosen file name to identify the models being saved, which are exported in the same order as they were loaded, edited, created, and/or discovered.

Part-Whole Partial Least Squares

In part-whole PLS analysis, a subset of variables is regressed onto the full set to which it belongs, with the objective to detect features other than the subset's variables that covary with them. In other words, the main aim is to determine whether a putative module is actually embedded within a larger module or subspace. Additionally, the technique is useful to visualize intra- and inter-modular variation, which can lead to the postulation of alternative module boundaries, or even alternative modules.

Consider, for instance, a structure that is partitioned to define four modules, namely {A}, {B}, {C}, and {D}. A PLS regression of, say, partition {A} onto the full, intact structure {A+…+D} will normally produce orthogonal paired vectors of {A} and {A+…+D} which account for decreasing portions of the covariance between the two sets. The first PLS axis, for instance, will account for most of the covariation between the part and the whole, and it is naturally expected to show the covariation of {A} as a partition and of {A} as an embedded subspace within {A+…+D}, as it would be similar to regressing a variable onto itself. The interest, however, is not on this expected association, but instead focuses on what other variables, from partitions {B}, {C}, and {D}, also covary with variables in partition {A}. For instance, a PLS axis showing covariation of both {A} and {B} when {A} is regressed onto {A+…+D} would suggest that maybe {A} and {B} comprise a single module instead of two. Furthermore, this regression could also reveal an association between {A} and only some of the variables in other partitions, or between part of {A} and parts of other partitions, which could lead to a complete reformulation of the boundaries of the putative modules which could be resubmitted to the GoF tests to compete with previously defined models, as shown in Márquez (2008).

A potential caveat of this method is that a part-whole regression will tend to preferentially show covariation between a partition within the whole structure and the partition in isolation. Therefore, a situation where only part of a putative module is actually an integrated module may be hard to detect in this analysis. Using the same example as above, this means that even if only part of {A} covaries with variables in other partitions, say {B}, the fact that {A} is regressed onto a dataset containing {A} may tend to favorably show covariation of *all* of {A} variables with {B}. Likewise, variables in {A} uncorrelated to {B} may obscure any relationship between {A} and {B} in favor of a stronger integration within {A}. While these situations should produce a clear signal in PLS axes beyond the first, a strategy that can be used to investigate the integrity of putative modules consists on corroborating that associations among partitions are observed irrespective of which of these partitions is regressed on the whole structure. For example, if the PLS 1 axis shows an association between partition {A} and part of partition {B} when regressing {A} onto {A+…+D}, then regressing {B} onto {A+…+D} should also show covariation between {A} and {B}, even if this time the association is between all of {B} and part of {A}. In many cases, it is reasonable to expect that the partial associations are evinced in both regressions, although whether they appear associated to PLS 1 or higher-order axes may depend on the strength of the covariation of a partition's part with respect to the overall integration within the entire partition.

In PLS, a measure of "relevance" of an individual axis is given by *singular values* λ_i, which measure the amount of variation in a dataset accounted for by the other dataset that is captured by the i^{th} PLS axis. The proportion of squared covariance between part and whole accounted for by a PLS axis is computed in Mint as $\lambda_i^2 / \sum \lambda_i^2$ (Rohlf and Corti, 2000). To test whether a singular value is larger than expected by chance, Mint implements a permutation procedure (Rohlf and Corti, 2000), in which PLS vectors and singular values are re-computed between the part dataset and random permutations of individuals (rows) in the whole dataset. A significant PLS axis is thus the one that accounts for a larger proportion of the squared covariance than obtained from the random permutations.

In Mint, selecting the **PART-WHOLE** PLS tool results in the list element to be populated with all currently loaded models, with the exception of model #1 (null model), as a "part" in this case is not actually a partition. As mentioned above, when using landmark data, partitions in PLS analyses should have at least three landmarks or Mint will produce an error message. The first step in a PLS analysis is to select a model, after which the list element and plot will show the individual modules postulated by this model and a **SELECT PART** tool will become available. At this point it is also possible to change the model selection by clicking on the appropriate tool. Selecting a part will cause Mint to carry out a PLS regression of this part onto the whole dataset, while at the same time allowing to change the selected part. The image shown in the plot area will show the variation implied by PLS 1, whereas the list element will allow selecting any of the available PLS axes. When using non-landmark data, the plot will show PLS vector coefficients, whereas when using landmark data the plot will show the deformation implied by the selected axis (with or without a TPS grid, as selected using the provided button). Finally, clicking on **SHOW PART AXES** button allows swapping between PART and WHOLE results.

After using the **PERMUTATION TEST** task to test for the significance of individual PLS axes, a text summary of these and general PLS tests is displayed in the status box for models currently selected in the list element.

Lastly, choosing the **SAVE RESULTS** task displays a list of available outputs, described as follows:

- **SAVE PLS VECTORS OF WHOLE DATASET** produces a matrix of coefficients of the PLS axes obtained for the whole-structure dataset, sorted from larger to smaller singular values. Note that in the case of landmark data, the number of rows equals the dimensionality of the shape of the whole structure—twice the number of landmarks minus four. The number of PLS axes (columns) equals the dimensionality of the smallest dataset (i.e., part data).
- **SAVE PLS SCORES OF WHOLE DATASET** produces a matrix of individual scores on each PLS axis.
- **SAVE PLS VECTORS OF PART DATASET** produces a matrix of PLS axis coefficients for the part selected for analysis, similar to the one produced for the whole dataset.
- **SAVE PLS SCORES OF PART DATASET** produces a matrix of individual scores of part data on PLS axes, similar to the one produced for the whole dataset.
- **SAVE SINGULAR VALUES AND PROPORTIONS OF CAPTURED COVARIANCE** produces a two- or four-column matrix containing the singular values of each PLS axis (rows) in the first column, the proportion of squared covariance accounted for by each PLS axis in the second column, and, if permutation tests have been run, *P*-values for the null hypothesis that singular values account for no more covariance than expected by chance in the third column and *P*-values for the null hypothesis that cumulative covariance accounted for by successive PLS axes is no larger than expected by chance in the fourth column.
- **SAVE PLS ANGLES AND CORRELATIONS** produces a two- or three-column matrix containing the angles in degrees between each pair of part and whole PLS axes (rows) in

the first column, the Pearson correlation coefficient between part and whole PLS axes in the second column, and, if permutation tests have been run, *P*-values for the null hypothesis that this correlation coefficient is no larger than expected by chance in the third column.

A worked example

Using the dataset and model files included with this release (`jaw.dat` and `jaws.model.batch12.txt`, respectively), a part-whole PLS analysis is carried out to demonstrate its main functionalities and applications. After clicking on the **PART-WHOLE PLS** tool button, we proceed by selecting a model which will provide the part(s) to regress onto the full structure. As done in Márquez (2008), the chosen hypothesis will be model #4, because this is the one in which the mandible has been partitioned into its six developmental modules.

First, let us regress module #1 (anterior portion of the incisor alveolus) as the "part" in the part-whole regression. The TPS plot Mint produces should look like this:

whereas the text output informs us that:

Currently selected PLS axis (#1) captures 56.0388% of the squared covariance between PART and WHOLE datasets. Correlation between the PLS vectors of these datasets is 0.91508, which corresponds to an angle of 23.7829 degrees. Run permutation tests to assess the significance of these values. Currently selected: MODEL #4, MODULE/PART #1.

As expected, there is ample similarity between the aspect of the whole structure captured by PLS 1 and the selected part, a result that is corroborated when looking at the TPS plot for the part (by clicking on **SHOW PART AXES**):

which shows a similar pattern of variation to the one observed for the *same landmarks in the whole structure*. When interpreting these results, we note that not all of the landmarks in partition #1 show the same pattern of intra- or inter-partition covariation, which may reveal lack of integration within the putative module or, more likely, absence of enough variation in some of its semi-landmarks. More interestingly, from the whole-structure plot we note that some of the landmarks that do not belong in partition #1 seem to covary along PLS 1. Specifically, the upper edge of the molar alveolus, part of the angular process, and the condylar process seem to "respond" to the regression in a similar way as the incisor does, thus suggesting an potential association among these regions.

In this comparison, PLS 1 accounts for about 56% of the squared covariance between part and whole, indicating that other axes may contain relevant information. Thus, PLS 2, which accounts for 17% of the variance, produces the following deformation:

in which, once again, an association between incisor, molar, and condyloid process is suggested, along with covariation with other regions, such as the rostral and caudal ends of the masseter insertion. Before advancing any further, though, it seems appropriate to test for the significance of singular values and correlations among PLS axes by carrying out a permutation test. Permutation tests in PLS analyses usually consist on randomizing one of the datasets to simulate the absence of association between both datasets (Rohlf and Corti, 2000). In Mint, specimens in the whole-structure dataset are randomly permuted to compute (1) the probability that the proportion of the squared covariance between part and whole datasets that is accounted

for by individual PLS axes is no larger than expected by chance, (2) the probability that the proportion of this squared covariance that is accounted for cumulatively by consecutive PLS axes is no larger than expected by chance, and (3) the probability that the correlation coefficient between PLS vectors of part- and whole-structure datasets is no higher than expected by chance.

In the *Sigmodontomys* dataset, a run of 1000 permutations to test for significance of regressing the incisor module onto the whole mandible produces the following text output for PLS axis 1:

Based on 1000 permutations, the observed percentage of squared covariance accounted for by PLS axis 1 (=56.0388%) is no larger than expected by chance with a P = 0.058. Cumulative covariance accounted for by PLS axes 1 through 1 equals 56.0388%, with a P = 0.058. The observed correlation between PART and WHOLE vectors in PLS axis 1 (=0.91508) is no higher than expected by chance with a P =0.001. Currently selected: MODEL #4, PART/MODULE #1.

Thus, even though that, as expected, part and whole are highly correlated on PLS 1, this axis accounts only for 56% of the (squared) covariance between the datasets, which does not exceed the proportion of covariance accounted for by PLS 1 in about 6% of the randomly permuted datasets. Even though non-significant results like this can be often interpreted as indicative of absence of a clear association between the datasets being compared, such interpretation is not an option in the present case, as the datasets are partly redundant, so we know that they do covary. A more appropriate interpretation for this result is that the association between part and whole in this case is not one-dimensional, and thus it is not appropriately summarized by the 56% of the covariance implied by PLS 1. One way to estimate the number of axes necessary to account for a significant portion of the covariance between the datasets is by testing the hypothesis that the first *k* PLS axes account for more covariance than expected by chance. In this test, a criterion for choosing the number of axes to interpret as those capturing a significant portion of the part-whole interaction could be the *minimum number of PLS axes for which a significant* P-*value is obtained from permutation tests*. In other words, we use the minimum dimensionality of a dataset for which a distinctly non-random portion of information is discernible.

Back to the incisor module, we notice that to get a $P < 0.05$, we must choose the first three PLS axes. The text output of the latter reads:

Based on 1000 permutations, the observed percentage of squared covariance accounted for by PLS axis 3 (=10.6797%) is no larger than expected by chance with a P = 0.671. Cumulative covariance accounted for by PLS axes 1 through 3 equals 83.4256%, with a P = 0.018. The observed correlation between PART and WHOLE vectors in PLS axis 3 (=0.81926) is no higher than expected by chance with a P =0.001. Currently selected: MODEL #4, PART/MODULE #1.

which leads us to conclude that the association between incisor and whole-mandible is at least three-dimensional. Adding further axes to this space will continue to show significant results,

but those could be accounted for, parsimoniously, by the significance obtained in the first three dimensions.

As the above text describes, PLS axes 1 through 3 account for 83% of the squared covariance between part and whole. For the plot for PLS 3, we get:

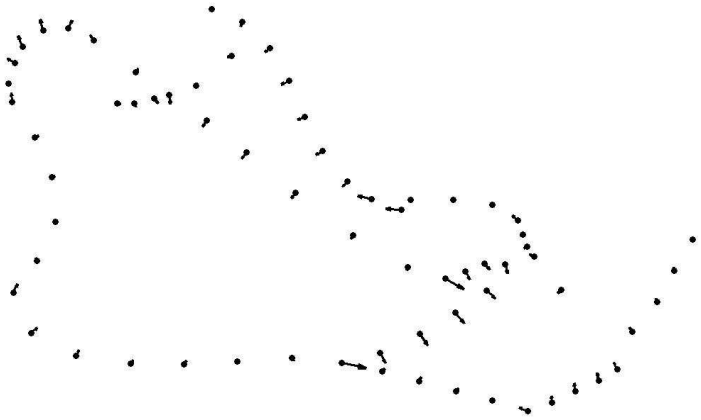
which suggests an association between the posterior landmarks in the incisor alveolus, the anterior portion of the coronoid process, the ascending portion of the incisor alveolus (also insertion site for masseters), molar alveolus, and, to a lesser extent, the angular process.

Summarizing previous results, PLS analyses have suggested a relationship between the anterior portion of the incisor alveolus and: (a) molar alveolus (PLS 1, 2, and 3), (b) condyloid process (PLS 1 and 2), (c) ascending portion of incisor alveolus/masseters insertion (PLS 1, 2, and 3), (d) anterior margin of coronoid process (PLS 3), and to a lesser extent, (e) angular process (PLS 1 and 3). Among these, the most consistent associations are (a) and (c), so we will look into those more carefully.

First, we will repeat the PLS analysis using the same model, this time regressing the molar alveolus and the ascending portion of the incisor alveolus (putative modules #2 and #4, respectively). We do this to verify whether there is an association between the incisor module and these modules, because the pattern observed above could have been caused by only part of the incisor module being associated to parts of the other modules.

When regressing the molar alveolus, as defined in model #4, onto the whole structure, we also find a three-dimensional solution ($P = 0.02$), whereby the most salient pattern is the high self-correlation within the structure. Additionally, plots reveal associations between molar alveolus and: (a) condyloid process (PLS 1 and 3), (b) angular process (PLS 1, 2, and 3), and (c) coronoid process (PLS 2). Although a rather weak association is noted between molar and incisor in PLS 2, the most relevant aspect of this interaction is observed in PLS 3, in which the molar alveolus seems to covary largely with *two* landmarks of the incisor module. Since these are also the most variable landmarks of the incisor module, it appears that the association observed above is the result of either a poor integration within the incisor module as defined by model #4, or the presence of additional factors integrating these structures. Both of these scenarios can be turned into Mint hypotheses and tested alongside other models.

Focusing for this example only on the incisor, we turn to regressing the ascending (ramal) portion of the incisor alveolus (module #4 of model #4), also insertion site for masseters, onto the whole mandible. In this case, PLS 1 accounts for 45% of the squared covariance between whole and part, for a $P = 0.01$. We can thus focus on this axis for this comparison, as PLS 2, even though it accounts for 20% of the covariance, the probability that this value exceeds the expectation from chance is $P = 0.47$. The deformation of the whole mandible implied by PLS 1 is:

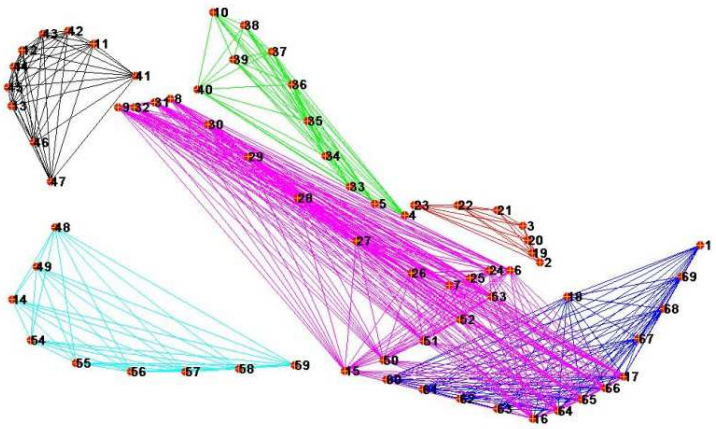


Besides revealing substantial self-covariation, PLS 1 suggests an association between the ramal portion of the incisor alveolus and (a) the anterior portion of the alveolus, (b) the anterior margin of the coronoid process, (c) the condyloid process, and to a lesser extent, (d) the angular process.

Although a full enumeration of the features suggested by these results lies beyond the scope of this guide, we can summarize some of their most consistent features and attempt to build a heuristic model based on them.

First, there is evidence of integration between the anterior and ramal portion of the incisor alveolus, although it may not involve all of the landmarks/regions comprising these putative modules. To model this, module #4 can be extended to include only the five landmarks and semi-landmarks in the mental area of the jaw (see graphic below), while keeping the anterior incisor module intact. Second, we find little evidence of strong integration between incisor and molar alveoli, so we leave them as independent. Although further regressions would show a more complex pattern of integration involving the condyloid, angular, and coronoid process, only the aforementioned modification to model #4 is demonstrated here for illustrative purposes.

Using the **MODEL EDITING** tool to modify module #4 of model #4 to include the incisor alveolus landmarks and semi-landmarks mentioned above (i.e., landmarks 16, 17, 64, 65, and 66), we get the following model (#14 after appending it to ongoing analyses):



(note how the blue, #1, and purple, #4 models overlap). Testing all 14 models using the γ*statistic with 1000 Monte Carlo replicates shows that the new model is not significantly different from the null expectation. Also, while it improves over most of the previous models, it still ranks second behind model #4, with γ* = -0.2888 (compared to -0.2958 for model #4) and jackknife support of 0.951for this rank (plus 0.049 for rank #1).

Iteration of this process thus can be used both to discover modular associations and to revise the boundaries of existing modules. In Márquez (2008: Fig. 6), the preferred model includes eight modules, at least one of which corresponds to a region with too little variation to be associated to any other module. The entire incisor alveolus in this model is not regarded as an integrated module, but as contributing to the ramus and molar alveolus and only partially statistically independent from other mandible regions.

Miscellanea

Saving plots

Anything shown in Mint's plot area can be saved as an image document from the **PLOT CONTROLS** menu. Mint offers two general formats for saving images—graphic bitmaps and vector graphics. Available bitmaps formats are 24-bit JPEG, 24-bit compressed TIFF, 24-bit uncompressed TIFF, and Windows Bitmap. Available vector graphics formats are PDF, EPS Level I/Color, EPS Level I/B&W, and Windows Enhanced Metafile.

TPS plot options

Thin-plate spline plots of PLS axes of landmark data can be adjusted to enhance visualization using four parameters, accessible from the **PLOT CONTROLS** menu:

- *Deformation magnification*: controls the factor by which shape variation is multiplied along the axis being shown. Default value is 1, which shows the implied deformation between the positive and negative values of the maximum absolute score along the axis shown. Allowed range is -5 through +5, with negative values reversing the direction of the deformation. A value of zero simply shows the Procrustes mean of the loaded dataset.

- *Grid extent*: controls the graphical overlap between the grid and the Procrustes mean upon which deformations are shown. Setting it to 1 means that the grid fits tightly around the mean, whereas larger values mean that the grid occupies a larger plot space than the mean does. Default value is 1.1 (i.e., grid occupies approximately 110% of the plot space occupied by the Procrustes mean).
- *Grid density*: controls the density of cells with which a deformation grid is depicted by assigning the number of cells on a side of the grid. Default value is 40 (i.e., 40 cells will be drawn along the long edge of the grid).
- *Vector magnification*: controls the factor by which deformation vectors are drawn on the Procrustes mean. It differs from deformation magnification (above) in that only vector length is altered, artificially, whereas the deformation per se is left unchanged. This is only useful for visualization purposes, as it does not produce accurate patterns of variation. Default value is 0, which corresponds to the original magnification; values above and below 1 shrink and magnify vectors, respectively.

Cited literature

Anderson, T. W. Asymptotic theory for principal component analysis. Ann. Math. Stat. 34:122-148.

Dietz, E. J. 1983. Permutation tests for association between two distance matrices. Syst. Zool. 32:21-26.

Márquez, E. J. 2008. A statistical framework for testing modularity in multidimensional data. *Evolution* 62:2688-2708.

Parsons, K.J., Márquez, E.J., Albertson, R.C. 2012. Constraint and opportunity: the genetic basis and evolution of modularity in the cichlid mandible. *The American Naturalist* 179:64-78.

Richtsmeier, J. T., S. R. Lele, and T. M. Cole, III. 2005. Landmark morphometrics and the analysis of variation. Pp. 49-69 *in* B. Hallgrímsson and B. K. Hall, eds. Variation: a central concept in biology. Elsevier, Amsterdam.

Rohlf, F. J., and M. Corti. 2000. Use of two-block partial least-squares to study covariation in shape. Syst. Biol. 49:740-753.

Walker, J. A. 2000. Ability of geometric morphometric methods to estimate a known covariance matrix. Syst. Biol. 49:686-696.

Zelditch, M. L., J. Mezey, H. D. Sheets, B. L. Lundrigan, and T. Garland. 2006. Developmental regulation of skull morphology II: ontogenetic dynamics of covariance. Evol. Dev. 8:46-60.