FURTHER TECHNICAL DETAILS of PHYLO.GLM
(Version 1.02)

by Alan Grafen

This file describes the use of macros and vectors. It has six sections:

1) MACROS AND VECTORS WHICH MUST BE SET BY THE USER

2) USER-AMENDABLE MACROS AND VECTORS

3) MACROS INTENDED FOR CALLING BY THE USER

4) SCHEMA OF THE PATTERN OF MACRO CALLS

5) CODE-CONTAINING MACROS

6) OTHER IDENTIFIERS

Sections 1 to 3 may of interest for reference. Sections 4 and 5 are purely technical, and are irrelevant to use of the program. They may give would-be re-programmers of the phylogenetic regression a sense of the complexity of the task, and will be essential for foolhardy tinkerers with the program. Section 6 contains details of other identifiers, including left-over vectors, and SC__. SC__ is a vector used internally in the program for storing various miscellaneous scalars - and each of these is explained. They include channel numbers; default settings for the search method used to find the optimal ρ; various pieces of information about the most recent analysis; and numerical tolerances, which I now explain.

There are places in the program where it really matters whether a number is zero or not zero. In order to prevent rounding errors from hiding a real zero, a numerical tolerance is defined such that any number less in absolute value than the tolerance is set to zero. In both cases, the tolerance has been set to 1e-6. This means, for example, that dividing an x-variable by ten repeatedly will eventually lead its being treated as zero, roughly speaking when the differences between elements are less than 1e-6. Biologically significant differences between data values in both x and y variables should be maintained healthily above 1e-6, therefore.

MACROS AND VECTORS WHICH MUST BE SET BY THE USER

Macro	Use
yv_	Contains the name of the y-variable for the next analysis.
c_	Contains the names of up to nine x-variables to be controlled for in the next analysis. See con_ in “User-amendable macros” below about controlling for more x-variables.
t_	Contains the names of up to nine x-variables to be tested for in the next analysis. See tst_ in “User-amendable macros” below about testing for more x-variables.
tx_	Needs to be set if using the taxonomic levels method of describing the phylogeny. It should contain the names of the vectors containing the taxa coded in them, with the highest level first, and the lowest level last. See tax_ in “User-amendable macros” below for dealing with more than nine taxonomic levels.
phy_	A vector containing the phylogeny, that must be set by the user either directly (by reading it in, for example) or indirectly (by calling mph_).

USER-AMENDABLE MACROS AND VECTORS

Macro	Default	Use
opt_	*various*	Controls output and user-interruptions. Details and defaults given in manual.
con_	c_	Contains up to nine names of macros, each of which can contain up to nine vector names. These vectors are controlled for in the analysis.
tst_	t_	Contains up to nine names of macros, each of which can contain up to nine vector names. These vectors are tested for in the analysis.
tax_	tx_	Contains up to nine names of macros, each of which can contain up to nine taxonomic levels vectors. The highest level macro should come first, and the lowest level last.
trn_	i w f h o	Contains the arguments of $TR for output “on”, applied in oon_.
trf_	*Empty*	Contains the arguments of $TR for output “off”, applied in oof_.

MACROS INTENDED FOR CALLING BY THE USER

Macro name	Arguments	Function
mph_	*None*	Transforms information about phylogeny from taxonomic levels vectors into a form the program can use. The information about the structure of the tree is stored in PHY_.
exf_	2	Creates the dummy variables for a categorical variable. The first argument is the categorical variable. The second is a macro containing the names to be used for the dummies.
ifc_	3	Creates the dummy variables for the interaction between a categorical variable and a continuous variable. The first argument is the categorical variable. The second is the continuous variable. The third is a macro containing the names to be used for the dummies.
iff_	3	Creates the dummy variables for the interaction between two categorical variables. The first two arguments are the categorical variables. The third is a macro containing the names of macros, which in turn contain the names to be used for the dummies.
dls_	*None*	Saves on user identifiers by deleting who_ exf_ ifc_ iff_ mph_ and other macros called by these internally. It is important to make any use of these macros before calling dls_. It will usually be sensible to delete dls_ itself after use.
go_	1	Performs an analysis with yv_ as y-variable, controlling for con_ and testing for tst_. A species is included if it has a ‘1’ in SPI_, and has no missing values in any vector in yv_, con_ or tst_.

SCHEMA OF THE PATTERN OF MACRO CALLS

This section is entirely technical and should never be needed in normal use of the program.

```
         |cou_                                   |cou_
         |                                       |psh_
         |                   |wk__      und_     |shr_
         |                   |                   |pll_
         |ou__      in__     |                   |ext_
         |                   |                   |cmb_
         |                   |
         |                   |wk2_      vnd_
         |
GO_      |ou__&&
         |uw__
         |csl_
         |clv_
         |cpl_
         |
         |         |psh_
         |         |ou__&&
         |         |ou__&&
         |         |                             |tpr_
         |         |                   |ft2_     |ext_
         |         |                   |         |ou__&&
         |         |         |uft_     |         |ou__&&
         |         |         |         |
         |         |         |         |cr__
         |         |         |
         |         |fnd_     |         |tpr_
         |         |         |         |ext_
         |ft1_     |         |         |ou__&&
                   |         |         |ou__&&
                   |         |ft3_     |ou__&&
                   |                   |mq__
                   |                   |shr_
                   |                   |ou__&&
                   |                   |ou__&&
                   |pll_
                   |ou__&&
                   |ou__&&


         |ou__&&
MPH_     |cmb_
         |uw2_


EXF_      ef1_

IFC_      if1_

IFF_      if6_      if7_

DLS_
```

Each macro intended for calling by the user appears on the left hand side in upper case. To the right appear the macros which that macro can call internally, in a vertical list. The extent of lists with more than one member is indicated by a vertical bar. The corresponding lists appear to the right of each member of the first list, and so on. The appearance of a macro in a list means it is called with a $WHILE, $USE or $SWI, and so may be invoked zero or more times. The && after all but the first mention of ou__ indicates that the macros called by ou__ are not repeated at each occurrence, and reference should be made to ou__'s first mention for the macros called. Calls of ou__ are always adjacent to a call of one of the macros ultimately callable by ou__ (cou_ to cmb_): it is the adjacent macro that is called in that invocation.

CODE-CONTAINING MACROS

A list of all macros containing GLIM code, detailing the macros they call and the function they perform. (In the order in which they appear in the program.)

<u>This section is entirely technical and should never be needed in normal use of the program.</u>

Name of macro	Macros it calls	Function of macro
csl_		Creates path segment lengths according to the “Figure 2” method of the source paper.
clv_		Creates path segment lengths using the taxonomic levels method.
cpl_		Creates path segment lengths using user-supplied node heights.
tpr_		Calculates the power-transformed path segment lengths, and the rho-terms in the likelihood.
ef1_		Calculates the dummy variable for one level of a factor.
exf_	ef1_	Calculates the dummy variables for a factor, by calling ef1_ for each.
if1_		Calculates the dummy variable for the interaction of a continuous variable and a factor.
ifc_	if1_	Calculates the dummy variables for the interaction of continuous variable and a factor, by calling if1_ for each.
iff_	if6_	Calculates the dummy variables for the interaction of two factors. It calls if6_ for each element in the macro containing the names of the macros containing the names of the dummies.

if6_	if7_	Calculates the dummy variables in one macro (for a factor by factor interaction), by calling if7_ for each dummy.
if7_		Calculates the dummy variable for the interaction between two factors (after checking that the end has not come).
go_	cou_ ou__ ou__ uw__ csl_ clv_ cpl_ ft1_	Performs an analysis using the current values of YV_, CON_, TST_ and SPI_. cou_, ou__, ou__ counts for storage and works out which species need to be dropped for missing values. uw__ amends the phylogeny to drop the missing species. One of csl_, clv_ and cpl_ is used to create path segment lengths. Then ft1_ is called to do the analysis. go_ also sets up all the basic facility vectors used by the other macros, and deletes unnecessary vectors when it starts, and when it ends.
mq__	rom_	Uses the residuals from the standard regression just performed to create the linear contrasts (qrs_) with which to convert the long to the short regression. If any linear contrast is identically zero, then rom_ is called to inform the user.
rom_		Calculates how many radiations are omitted from the short regression, and prints out how many, and which nodes they are.
shr_		Puts into the first section of a vector its short regression values
ou__	in__	Calls in__ repeatedly with different values of %z1. The arguments of in__ are the contents of ou__'s own %1, filled out with the dummy macro SHM_.
in__	wk__ wk2_	Calls wk__ and wk2_ for each of their arguments in turn. Their arguments are the vectors named in the macro which is in__'s %z3th argument.

wk__	und_	Calls und_ with appropriate argument.
und_	cou_ psh_ shr_ pll_ ext_ cmb_	Switches to one of a number of possible macros according to the value of %z5. This along with the action of wk2_ is the ultimate point of the ou__, in__ double act; ou__ and in__ are just administration.
wk2_	vnd_	Either adds or doesn’t add to the current fit the vectors which are its arguments, depending on the setting of %z6.
ft1_	psh_ ou__ ou__ fnd_ pll_ ou__ ou__	Stores the species values of the required vectors (psh_ ou__ ou__), performs the analysis (fnd_), and then restores the vectors to their original state (pll_ ou__ ou__)
ft2_	tpr_ ext_ ou__ ou__	Using the current value of rho (in gr__(%z4)), calculates the transformed path segment lengths and consequent long regression weights (tpr_), then transforms the vectors to their long regression values and fits the model (ext_ ou__ ou__). The purpose of ft2_ is to calculate the likelihood for a given value of rho.
ft3_	tpr_ ext_ ou__ ou__ upt_ ou__ upt_ mq__ shr_ ou__ upt_ ou__ upt_	Once the maximum likelihood estimate of rho has been found, ft3_ performs the actual analyses. It creates path segment lengths (tpr_), extend the vectors and fit the long regression on control and test (ext_ ou__ ou__), and then allows user intervention at Place 2 (upt_). It refits just the control variables (ou__) and allows user intervention at Place 1 (upt_). It reports on missing denominator degrees of freedom (mq__). It shrinks and fits the control variables (shr_ ou__) and allows user intervention at Place 3 (upt_). It shrinks and fits the test variables, and allows user intervention at Place 4 (upt_).

fnd_	uft_ ft3_	If OPT_(17)=0, calls uft_ until the maximum likelihood estimate of rho is found. Then calls ft3_ to perform the definitive analyses. fnd_ then informs if there is any loss in numerator degrees of freedom. This is done by comparing the degrees of freedom of the long (ZE__) and short (ZF__) regressions.
uft_	ft2_ cr__	Performs one gridful of interations in the search for the maximum likelihood value of rho. Calls ft2_ to fill in the missing likelihoods in the current grid of rho values (gr__). Uses the position of the maximum to create the new grid. For a maximum at an edge of the grid, cr__ has to be called to create the new grid.
cr__		Called when the maximum likelihood occurs at the edge of the grid (gr__). Moves the two best values of rho three places sideways, and completes the grid in geometric progression.
cou_		Just adds one to %z3 when called.
vnd_		Adds all its arguments to the current model when called. Its arguments are the contents of one of the macros named in CON_ or TST_, filled out to %9 with ZO__, a vector of zeroes.
psh_		Stores in STO_ the top sc__(1) values of its %z2nd argument, and extends it to length sc__(2) by padding with zeroes.
pll_		Replaces in the top sc__(1) values of its %z2nd argument the original values held in STO_, and cuts off the rest of the vector.
ext_		ReSTOres the species values of a vector (its %z2nd argument), calculates the values of its higher node section, and then expresses ALL values as differences from the parent node's mean. If OPT_(16)=1, prints the mean of each variable as it restores it.

mph_	ou__ cmb_ uw2_	Creates the phylogenetic vector phy_ from taxonomic level vectors held in macros whose names are held in tax_. Works by applying cmb_ to each taxonomic level vector (ou__), then applying cmb_ to a created species vector (=%cu(1)). uw2_ then unwrinkles the phylogeny, deleting single-daughtered nodes. In parallel it records in hst_ the level of each node. Also records the number of taxonomic level vectors in sc__(16) to allow checking of the length of the heights vector prior to clv_
cmb_		The in__nermost macro in mph_. It adds a lower level vector's information to a working version that includes all higher vectors' information. The information is on further phylogenetic splits, and on the level at which they occur.
uw2_		Unwrinkles a phylogeny, that is, excises single daughter nodes and renumbers. Amends the levels vector hst_ in parallel.
uw__		Takes the permanent phylogeny phy_, and creates a phylogeny txp_ appropriate for the current analysis. Species are omitted according to the vector spu_, and higher nodes need consequent rejiggling.
who_		Identifies the higher nodes of phy_ by creating one vector with an included species, and another with a just-excluded species, for each node.
oon_		Switches output "on", which means directs output to channel number SC__(4) and sends to the transcript file the information detailed in the macro TRN_. SC__(4) is by default the current output channel when PHYLO.GLM is read in. TRN_ by default is set to "i w f h o".

oof_

Switches output "off", which means directs output to channel number SC__(5) and sends to the transcript file the information detailed in the macro TRF_. SC__(5) is zero by default, causing no output. TRN_ by default is set to " ", i.e. no transcript.

upt_

This macro is called to $SUSpend the program’s execution and return control to the user temporarily. It provides optional dire warnings and advice, and sets a flag (SC__(22)=1) to indicate that an interruption is in progress.

upt_

Allows a user-interrupt, first setting a flag (SC__(22)) so that GO_ can bounce re-entrants. Prints dire warnings and advice, which can be all but suppressed by setting OPT_(20). Whether upt_ is called at Places 1 to 4 depends on the settings of OPT_(21) to OPT_(24).

dls_

Deletes unneeded identifiers to save space, but at the cost of preventing i) re-creation of the phylogeny (mph_) ii) construction of further factors and interactions (exf_, iff_, ifc_) and iii) identification of higher nodes (who_). So do all these things before using dls_.

OTHER IDENTIFIERS

Identifier	Use
shm_	A macro containing just “SH__ ”. shm_ is used to fill all the arguments of a macro, to be overriden by the unknown number of macros contained in #CON_, #TST_ and so on.
sh__	A vector of length 1. It is used in shm_ and elsewhere as a place-holder. The program knows that it has reached the end of the list of “real” arguments by the value of %CU(arg==arg), which is 1 for sh__ and not 1 otherwise.
spu_	A vector of species length, containing 1 for species included in the most recent analysis.
hst_	A vector recording the heights of the levels of the nodes in the corresponding elements of phy_.
on_	A vector containing the names in the original phylogeny, as used in phy_, of the nodes corresponding to units in the long regression.
b_	A vector containing the actual node heights used in the last analysis, before rho-transformation.
qrs_	A vector containing the linear contrasts used to form the short from the long regression.
wl__	The weights vector in the long regression
wf__	A vector containing the averaging coefficients for calculating the averages at higher nodes from the species data.
sc__	sc__ holds 23 miscellaneous values, each of which is explained separately. A star (*) indicates those that can sensibly be altered by the user - the others are automatically computed for the current analysis, and need not and should not be altered.
sc__(1)	The number of species in the dataset.

sc__(2) The length of vectors in the long regression. Includes missing species, but excludes higher nodes in the original phylogeny that do not exist in the phylogeny for included species.

sc__(3) The number of datapoints in the short regression, including those that are omitted for lacking a phylogenetic degree of freedom.

sc__(4)* The channel number for output. Set to the current output channel (%coc) when PHYLO.GLM is read in.

sc__(5)* The channel number for output when output is “switched off”. Zero by default.

sc__(6) Not used (was in simulations).

sc__(7) Stores the rho-term of the likelihood.

sc__(8)* Specified accuracy for the fitting of rho. By default is 0.02.

sc__(9) Contains a lower bound to the actual accuracy in the fitting of rho.

sc__(10)* The number of additionally fitted parameters, for subtraction from denominator DF. It is by default set to 1 (for rho). If an a priori rho is used, sc__(10) should be set to zero.

sc__(11) The number of non-omitted species.

sc__(12) The number of included datapoints in the long regression.

sc__(13) The number of included datapoints in the short regression.

sc__(14)* The value of rho below which the search ceases. Set by default to 0.002.

sc__(15) Indicator for type of node heights. 1=“Fig 2”, 2=“Taxonomic levels”, 3=“Complete specification”.

sc__(16) Set by mph_ as the number of levels in the taxonomy. -1 if unset.

sc__(17) The number of test variables

sc__(18) The number of variables needing storage

sc__(19) The length of PHY_, the vector containing the full phylogeny for all species. Recalculated at each use of GO_.

sc__(20)* The tolerance for elements of qrs_ to count as zero. By default is 1e-6.

sc__(21) The grand mean of the y-variable

sc__(22) A flag set during a user-interruption, to allow GO_ to bounce re-entrants.

sc__(23)* The square of the tolerance for elements of any vector in the long regression to be zero. By default is 1e-12.

sc__(24)*, sc__(25)* Contain the lower (24) and upper (25) bounds of the initial search region for rho. The values by default are 0.1 and 0.5. The search is not restricted by these initial bounds. The user might save time by setting these more finely if experience shows that rho is usually within a narrower range. If rho is usually outside this range, it might be worth changing the bounds to include the likely values. sc__(24 and 25) are re-read at each use of GO_, and so can usefully be reset in between uses of GO_. Both values must be strictly greater than zero, and sc__(25)>sc__(24).

sc__(26) Tracks whether the program is in uft_ or ft3_, so that the means of the variables requested by OPT_(16)=1 are printed only during the final analysis of ft3_, and not at each iteration of the search.