TUTORIAL

Automatic Resonance Assignment with GARANT

Christian Bartels

Laboratoire de Chimie Biophysique

Institut Le Bel

Université Louis Pasteur

4, rue Blaise Pascal

67000 Strasbourg

France

Summary

By explaining the example macro *assignHom.gar,* it is described how to obtain resonance assignments for proteins with the program GARANT. The macro automatically determines the resonance assignment of the mutant protein Tendamistat(R19L) using peak lists from homonuclear COSY, TOCSY and NOESY spectra and the structures of the wild type protein. Changes necessary to adopt the macro for other proteins and other types of input data are indicated. It is advantagous, if the reader is familiar with the publications describing GARANT (Bartels, C, Billeter, M, Güntert, P & Wüthrich, K (1996). *Automated Sequence-specific NMR Assignment of Homologous Proteins using the program GARANT*. J. Biomol. NMR **7**, 207-213; and Bartels, C., Güntert, P., Billeter, M., and Wüthrich, K. (1997). *GARANT — a General Algorithm for Resonance Assignment of Multidimensional Nuclear Magnetic Resonance Spectra.* J. Comp. Chem. **18**(1), 139–149). Some additional details about the structure of the program (not its application) can be found in C. Bartels, *Methoden der Zuordnung mehrdimensionaler magnetischer Kernspinresonanzspektren zur Strukturbestimmung von Makromolekülen*, Dissertation ETH No. 10966, Zürich, 1995.

Running the Macro *assignHom.gar*

The macro is in the directory *.../examples/tend* together with the input data and an example of the output data. A convenient way to run the macro and redirect the output into a log file is to type in a UNIX shell

> garant *assignHom* <*name*> <*seed*> >&! *assignHom.out &*

where <*name*> has to be substituted with an identifier that is added to the names of all output files, and <*seed*> by an integer number that is used to initialize the random number generator. If you run the macro a second time with the same data, *e.g.*, to check the convergence of the calculations, you should use a different identifier and a different integer number. If GARANT is properly installed and the paths in the macro are properly set (see below), output similar to the one in *assignHom.out-reference* will be produced. Each time a resonance assignment is found which has a higher score (mutual information) than all previous resonance assignments it is written out, *i.e.*, the macro *writebest.gar* in the directory *.../macro* is called. This macro writes out lists with expected peaks (file names *E*TS<name>.peaks*), lists with assigned, observed peaks (file names **TS<name>.peaks*) and a log file (file name *TS<name>.log*) which describes the quality of the resonance assignment.

Description of the Macro *assignHom.gar*

The macro, first, defines the input data to be used. Then, the data structures describing the expected and observed peaks are generated. Finally, the general optimization macro *opt* is called, which does the actual optimization and needs not to be changed by the user.

It follows a short description of the different commands.

`parameter suffix rseed`

Defines that the parameters specified when calling the macro (*e.g.*, <*name*> and <*seed*>, see above) should be put into the variables *suffix* and *rseed.*

See also: help parameter

`var i`

Defines the variable i to be a local variable, i.e. to be valid only in this macro.

See also: help var

`set ...`

Defines the values of variables that are used later on in the macro or in the macros opt.gar and writebest.gar.

See also: set

`set name     = TS{$suffix}`

Defines the variable *name* which is added to the names of output files in the macro *writebest.gar*. The value of the variable *suffix* is substituted for *{$suffix}* by the command interpreter (see also: help variable).

`set dir      = /usr/people/cb/garant-1.0/examples/tend`

Defines the variable dir which is later used as path to access input data and write output data.

`set atomlist = tendmut`

Sets the variable atomlist to the file name of the XEASY proton list (see XEASY program manual) which is later used to write out assignments. The default file extension is *.prot* and does not need to be specified. Explanation: peak lists in the XEASY format contain assignments as a set of numbers. To translate these numbers into atom names a translation table, called proton or atom list, is required.

set sequence = tendmutDIANA

Defines the name of the file that contains the protein sequence in XEASY file format (see XEASY program manual). The default file extension is *.seq* and does not need to be specified.

```
set cosypl   = COSauto2
set tocsypl  = TOCauto1
set noesypl  = NOEauto2
```

Defines the names of the files that contain the lists with the positions of the observed peaks (peak lists) in the XEASY file format. The same names are used to generate file names for writing out the found assignments (see macro *writebest.gar*). The default file extension is *.peaks* and should not be specified (unless the macro *writebest.gar* is adapted).

set reftocsy = TendMutTOC_4

GARANT provides the possibility to compare resonance assignments. In the present example macro this possibility is used to provide (in the file TS<name>.log) a comparison between the correct manual assignment and the automatically determined assignments. In other applications this facility might be used to compare the results of two independent GARANT runs. Found discrepancies can point out the parts of the resonance assignment that could be wrong. The variable *reftocsy* should be set to the file name of an assigned TOCSY peak list which is to be used for the comparison. A good strategy for a *de novo* resonance assignment is to run the macro a first time without making a comparison, *i.e.*, changing the command to

```
set reftocsy =
```

In a second run the results from the first run can then be used for the comparison, *e.g.*, by setting

```
set reftocsy = TOCauto1_TSI.peaks
```

The following variables describe the data from the wild type protein and how it should be used for the resonance assignment:

```
set mutres      = 19
```

mutres is a list of all those residues which are different in the wild type and in the mutant protein. If there is more than one residue, separate them by blanks (see also: help variable).

```
set freq        =
```

freq is the name of the proton list (see also: XEASY manual) with the chemical shifts of the wild type protein. If, as it is now, no file name is specified, chemical shifts of the wild type protein are not used.

```
set stdDevHN    = 0.15
set stdDevAli   = 0.05
```

stdDevHN and *stdDevAli* are estimates for the average difference between the chemical shifts of the wild type and the mutant protein. They apply to the amide protons and aliphatic protons, respectively. To distinguish other types of chemical shifts, define suitable variables and add the adequate *define frequency tolerance* command (see also: below and help define frequency tolerance).

```
set strucs      = $dir/tendwild-
```

File name of the files containing the coordinates of the wild type structures in DIANA coordinate file format.

```
set nrStrucs    = 20
```

Number of conformers describing the wild type structure. Can be set to 1, if a X-Ray structure is used.

```
set limitdist  = 3
set limitdist2 = 4
set minStrucs  = 20
set minStrucs2 = 5
```

These are parameters used to generate likely NOE's from the structures of the wild type (see also: below and help make distance noe). For the present example, NOE's with a probability of being observed of 1 will be generated if the corresponding distance is smaller than 3 Å in all 20 wild type conformers, and NOE's with a probability of 0.2 will be generated if the distance is smaller than 4 Å in at least 5 conformers.

set savemacro = writebest

During the optimization, each time a resonance assignment is found which has a higher score (mutual information) than all previous resonance assignments, it is written out. For this purpose a macro is called. The variable *savemacro* is set to the name of this macro, e.g., in the present example, *writebest.gar* is called. This is a standard macro in the folder .../macro. If other types of spectra are to be used, another macro for writing out the resonance assignments has to be written (see also: example macros *.../examples/cyc/assignHet.gar* and *.../examples/cyc/writenoesys.gar*).

set sizePop = 100

This variable specifies the number of resonance assignments to be used by the genetic algorithm. With small values the optimization will run faster but the produced resonance assignment will be of lower quality. 100 is a good value for production runs. If you have a very fast computer you might set it to larger values. To test whether the macro is set up properly, runs with a population of 10 are appropriate. Don't expect any correct resonance assignments to be produced by these test runs.

spectrum COSY

Specifies that peaks from a COSY spectrum should be used. Other spectra types can be used (see help spectra types)

See also: help spectrum

set COSY_acc_w1 = 0.02
set COSY_acc_w2 = 0.02

Specifies the accuracy of peak positions in the ω1 and ω2 dimensions of the COSY spectrum, respectively. For each spectrum and each dimension, the average position of all peaks assigned to the same atom is calculated. The specified value is the maximal allowed deviation in ppm of a peak from this average position.

See also: help spectra variables

load peaks COSY $dir/$cosypl

Positions of observed COSY peaks are read in from the file *$dir/$cosypl.* The file should be in the XEASY peak list file format (see XEASY manual).

See also: help load peaks

```
spectrum NOESY
   set NOESY_acc_w1 = 0.02
   set NOESY_acc_w2 = 0.02
   load peaks NOESY $dir/$noesypl
spectrum TOCSY
   set TOCSY_acc_w1 = 0.02
   set TOCSY_acc_w2 = 0.02
   load peaks TOCSY $dir/$tocsypl
```

Similarly, it is specified that NOESY and TOCSY spectra should be used. The accuracy of peak positions in these spectra is defined and the peak positions are read from the specified files.

```
set pick_acc      = 2.5
```

Specifies the maximal difference between chemical shifts of the same atom in different spectra. For each atom, an average chemical shift is calculated from the positions of all the observed peaks that are assigned to it. The maximal allowed difference between the position of a peak (e.g., in a COSY spectrum) assigned to a particular atom (e.g., in the ω1 dimension) and the average chemical shift of this atom is defined by

pick_acc * COSY_acc_w1.

The remaining part of the macro does neither depend on the protein which should be assigned nor on the types of the spectra that are used.

```
load sequence $dir/$sequence
```

The amino acid sequence of the protein (*e.g.*, mutant protein Tendamistat(R19L)) is read.

```
if ($def('reftocsy')) then
   load comparison sequence  $dir/$sequence
end if
```

If a comparison with an existing resonance assignment should be made, read in the amino acid sequence which was used for this assignment. In the present case, in which the comparison is made with the known resonance assignment, the amino acid sequence of Tendamistat(R19L) is read again. However, it would be possible to use another sequence, *e.g.*, in the case where spin system assignments have already been made and the assignment produced by GARANT should be compared to these spin system assignments.

pseudocorrection

The standard macro pseudocorrection.gar is executed which defines pseudo atom corrections of 1 Å for methyl groups. These corrections are applied when generating expected NOE's from the conformers of the wild type protein.

make coherences

Data structures for the atoms of the protein are initialized.

See also: help make coherences

make couplings

Expected scalar couplings are generated based on the covalent structure of the protein.

See also: help make couplings

```
set i=1
do
   if ('$mutres(i)' .eq. ' ') break
   make noe ${mutres(i)-1} ${mutres(i)+1}
   eval i = i+1
end do
```

For all mutated residues in the list *mutres* (see above), generate expected NOE's based on the covalent structure using some empirical rules. *E.g.*, intraresidual peaks, $d_{\alpha N}$ and d_{NN} peaks are generated for these residues. If no structure from a homologous protein is available these commands should be substituted by *make noe* (see also: example macro *.../examples/cyc/assignHet.gar*)

See also: help make noe, help do, help eval

```
do i 1 $nrStrucs
   if ($i.le.9)  then
      set temp = {$strucs}0{$i}.amfm
   else
      set temp = {$strucs}{$i}.amfm
   end if
   load coord $temp
end do
```

Read the coordinates of the wild type protein. The file names containing the coordinates are generated from the contents of the variable *strucs* (see above), a running index, 01, 02, ..., 20 used to distinguish the different conformers and the extension *.amfm*.

See also: help load coordinates, help variable

```
make distance noe floating $minStrucs  $limitdist  1.0
make distance noe floating $minStrucs2 $limitdist2 0.2
```

Expected NOE's are generated based on the loaded conformers using the criteria described above.

See also: make distance noe

```
make peaks
```

Expected peaks are generated from the already generated NOE's and scalar couplings.

```
if ($def('freq')) then
   load frequencies $dir/$freq
   define frequency tolerance *  $stdDevAli
   define frequency tolerance HN $stdDevHN
end if
```

If the chemical shifts from another protein should be used (variable *freq* is defined), they are read from the specified file and the expected deviations of these chemical shifts from those of the protein to be assigned are specified.

See also: above, help load frequencies, help define frequency tolerance, help syntax

```
if ($def('reftocsy')) then
         load   comparison   assignments   TOCSY   $dir/$ref
$dir/$atomlist 1.0
end if
```

If a comparison with an existing assignment should be made, the assignments stored in the peak list *$dir/$reftocsy* are read. The atom list *$dir/$atomlist* is used to translate the numbers which specify the assignments in the peak list into atom names. For each of the peak assignments, corresponding observed peaks, previously read in using the load peaks command (see above), are identified. The last parameter (1.0) specifies that for this identification the peak positions should be used with the tolerances defined above for each dimension of the TOCSY spectrum (variables *TOCSY_acc_w1* and *TOCSY_acc_w2*) multiplied by the parameter 1.0.

See also: help load comparison assignments, help load assignments

`defaultreport`

The standard macro defaultreport.gar is called (see below) to define the kind of report that is written each time a better resonance assignment is found during the optimization.

`seed random numbers $rseed`

The random number generator is initialized.

`opt $sizePop $savemacro`

The standard macro opt.gar is called to perform the optimization.

Description of the Macro defaultreport.gar

The report is defined, which is written out using the commands *report fragments* and *report coherences* by the macro *writebest.gar* each time a better resonance assignment is found. An example output can be found in the file *../examples/tend/TSI.log*. Reports can be produced either on the basis of individual residues (*report fragment* and *criteria fragment* commands) or on the basis of individual atoms (*report coherences* and *criteria coherence* commands). In general, it is recommended to use the former since the reports for individual atoms produce a huge amount of information that is itself difficult to analyze. A reasonably brief explanation of how to use the commands can be found in the help file on the *report fragment* and *report coherences* commands. More detailed information is given in the help files for the various *criteria* commands.

Description of the Macro writebest.gar

The current assignment which was generated by the optimization routine is written out. The macro is set up to work with homonuclear COSY, TOCSY and NOESY spectra. It has to be adapted if other spectra are to be used.

```
reset possible assignments
copyto possible assignments
```

Each observed peak has associated a list with assignment possibilities. These lists are emptied and the current assignment, generated by the optimization routine, is copied to these lists.

See also: help reset possible assignments and help copyto possible assignments

```
write peaks TOCSY $dir/{$tocsypl}_$name $dir/$atomlist
write peaks COSY  $dir/{$cosypl}_$name  $dir/$atomlist
write peaks NOESY $dir/{$noesypl}_$name $dir/$atomlist
```

The observed peaks are written out into the files *$dir/{$tocsypl}_$name, $dir/{$cosypl}_$name, $dir/{$noesypl}_$name* together with their assignments. The first assignment in the lists of assignment possibilities generated above is taken. To write out the entire lists, one could use the *write possible assignments* command. To translate atom names into atom numbers, the atom list *$dir/$atomlist* is used.

See also: help write peaks, help write possible assignments

```
write expected peaks TOCSY $dir/E{$tocsypl}_$name $dir/$atomlist
write expected peaks COSY  $dir/E{$cosypl}_$name  $dir/$atomlist
write expected peaks NOESY $dir/E{$noesypl}_$name $dir/$atomlist
```

All expected peaks, whose atoms have a defined chemical shift, are written out. The peaks are positioned at the chemical shifts of their atoms. To translate atom names into atom numbers, the atom list *$dir/$atomlist* is used.

See also: write expected peaks

```
report calculate
write log $dir/$name
   report global
   report fragments
close log
```

Reports are written to the file $dir/$name about the global quality of the assignment and about the assignments of each of the residues. An example for the produced output can be found in .../examples/tend/TSI.log. In the section describing the global quality, the mutual information is most important. The mutual information is related to the overall quality of the assignment, larger values correspond to better assignments.

See also: help report fragments, help criteria ...

Adapting the Macros for Different Systems

Different proteins. To assign other proteins using homonuclear COSY, TOCSY and NOESY spectra and information from homologous proteins, it is sufficient to make a copy of the example macro assignHom.gar and to properly set the variables (file names, number of structures to use, etc.) in the first part of the macro. If no homologous structure is available, replace the block of commands which generates NOE's from the conformers by the command *make noe* (see also: example macro *.../examples/cyc/assignHet.gar*).

Different types of spectra. To use other types of spectra changes have to be made to (copies of) the macros *assignHom.gar* and *writebest.gar*. Optionally you might want to change the report describing the resonance assignment by editing (a copy) of the macro *defaultreport.gar* or including suitable *criteria* ... commands in your copy of *assignHom.gar* (see also *.../examples/cyc/assignHet.gar*).

In the *assignHom.gar* macro, the new spectrum has to be invoked using the *spectrum* command, and the variables specifying the accuracy of peak positions and the order of the dimensions have to be set. An example of how to include a hetero nuclear 3D spectrum can be found in the macro *.../examples/cyc/assignHet.gar*. Take care to define the order of dimension correctly (see also help spectra variables and help spectra types).

In the *writebest.gar* macro, commands have to be added to write out the observed peaks and the expected peaks. An example can be found in the macro *.../examples/cyc/writenoesys.gar*.

To add new types of spectra to the source the files *map.C*, *specTypes.C* and *specTypes.h* have to adapted. You might use the UNIX command

```
grep N15NOESY *.[C,h]
```

to identify parts of the code in which different types of spectra are distinguished. The constant *NREXP* in *specTypes.h* has to be set to the number of defined types of spectra.

Folding. In general, it is assumed that peak positions read into GARANT correspond to peaks which are not folded. For a heteronuclear 3D spectrum in which some peaks are folded, it has been found to be a good strategy, to produce a peak list that contains all possible unfolded positions of these peaks, *e.g.*, using the corresponding heteronuclear 2D spectrum and the [uf] command of XEASY. Alternatively you might use the *unfold peaks* command of GARANT which can, however, not use the information from the corresponding heteronuclear 2D spectrum and might therefore generate a huge number of possible peak positions.

See also: help unfold peaks

Using known assignments. In general, it is recommended to use GARANT prior to any interactive analysis of the spectra (see below). Nonetheless, it is possible to use already known assignments. For this purpose you have to read the assignments form an assigned peak list with the command

```
load assignments
```

(see also: help load assignments). To prevent these assignments to be written out again, this has to be followed by the command

```
reset user assignments
```

for each of the spectra for which assignments were read. Subsequently you have to specify the atoms whose assignments should not be changed anymore by the program. *E.g.*, to specify that the assignments of all atoms of residue 36 to 45 and those of the HN and Hα atoms of residue 77 should be fixed add the following lines to your macro:

```
do i 36 45
   fix assignments "*($i)"
end do
fix assignments "HN(77)"
fix assignments "HA(77)"
```

Frequent Problems and Some Remarks

1) Prepare your input data carefully. The results you obtain with GARANT can only be as good as your input data. For example, all spectra have to be calibrated in the same way, they have to be recorded at the same temperature, strong noise bands should not be included as peaks in the peak list. Make a short test run (by setting *sizePop* = 10) and check that the lists with the expected peaks that are written out (file names E*.peaks) match approximately your spectra.

2) Even very good peak lists do not completely represent the recorde spectra (*e.g.*, peak shapes are missing). There is therefore a point at which improvements in the peak picking will have only a limited effect on the quality of the resulting resonance assignment. Produce a good peak list and use your time to check the resonance assignment proposed by GARANT rather than trying to generate perfect peak lists.

3) GARANT is not expected to assign spectra that can hardly be assigned interactively. Bear in mind that the program has less information available than can be used in interactive analysis of the spectra. GARANT works best, if there is redundant information. Use the program to produce initial resonance assignments which you then improve interactively. Don't expect GARANT to improve your interactive assignments.

4) The order of the dimensions of 3D spectra (see above) has to be specified correctly. Again, use a short test run to check that the produced output peak lists match your spectra, *e.g.*, proton shifts in proton dimensions, diagonal peaks are present, peaks are aligned in dimensions where you expect them to be aligned.

5) Peaks from both sides of the diagonal have to be included into the input peak list.

6) When executing a macro the suffix .gar should be omitted, *e.g.*, use

   garant assignHom ...

   and not

   garant assignHom.gar ...