Use of BioPack

updated: 12 May 2010

BioPack is a very large add-on package for VNMR, providing the best sequences available from Varian for use with proteins and rna/dna. BioPack is installed into the /vnmr (system) area on the INOVA-600. This installation has not been repeated on the INOVA-500, so the following will work *only* on the 600. And this is for VNMR 6.1c only (which is what is currently installed on the 600 and INOVA-500).

I. First-Time Setup of BioPack for Individual User Accounts:

To use BioPack the first time from your account, you should check that the following has been done. If a step seems not done or incomplete, the following steps can be redone at any time:

1. Copy the probe files into your account. Do the following in a UNIX terminal window:

   cp -R /vnmr/probes ~/vnmrsys

   Set your probe definition to the BioPack probe name, e.g., (from vnmr):

   probe='HCN5'

   HCN3 and HPX are the other two probes used on the I600.

2. Let BioPack backup (it will not delete) old BioPack or ProteinPack macros, sequences, parameter files, etc. The following also (I believe) sets up various required global parameters. Run the following at the VNMR command line:

   BPrmlocalfiles

   Note that any files you had that overlap with the new BioPack are backed up to:

   ~/vnmrsys/BioPack.dir/backups

3. Set the following parameter in vnmr:

   BPinstall=1

   If vnmr complains that BPinstall doesn't exist, do the following in vnmr:

   create('BPinstall','string','global')

   Then set BPinstall=1.

BioPack obtains customizations from the probe files, not from parameter files as ProteinPack used to. You should now be able to load in BioPack experiments without errors showing up. Try the following to test your setup (from vnmr):

ghn_co

II. BioPack Philosophy:

BioPack uses a probe-based philosophy. Whenever parameters are read in, or changed by a calibration, they come from a probe file. Calibrations will change the parameters in the probe file, but will backup the old probe file first (so it should be safe to play with the calibrations). The primary probe files are kept in /vnmr/probes.

The primary calibration buttons are under MAIN MENU → SETUP → PROTEINS.

A basic issue with BioPack, under the PROTEINS button anyway, is that it always assumes the following:

a doubly labeled sample, and in H2O:D2O (90:10 or close to)

I'm not certain what you can do with the calibrate buttons if those conditions are not met. I'm still not sure what the WATER calibration buttons do.

So if you are not doubly labeled, I suggest you do your best to get a good proton pw90 and tof, and go with that. The key for water suppression is still *absolutely* dependent on the quality of your shims.

Updating parameters without using the calibrate buttons:

BioPack bases manual parameter updates off the **ghn_co** parameter set. Suppose you have new parameters; doesn't really matter which, but suppose new **pw90 tof tpwr_cf** and **temp**. Do the follow to get these updates into the probe file, and to rebuild the shapefiles:

Load in the ghn_co parameter set: **ghn_co**

Change parameters: **pw pw90 tof tpwr_cf temp**

Always change *both* **pw** and **pw90**, as it is never completely clear which will be used.

Start the manual update macro: **BPbiopack2**

This process will take a minute or so to complete. It will update your probes file, and replace the shapes in the shapelib folder.

Fussing with calibration runs:

I’ve found it to be not uncommon to need to stop a calibration acquisition, change a parameter, and restart it. There are a number of reasons for this, but the most common is that a canned package just cannot accommodate all possibilities of samples, spectrometers, etc. So understanding how to do this is useful, and not difficult.

Suppose BioPack is calibrating **tof**, using a **tof:satfrq** array. You should observe a V in the intensity profile as the array is being acquired and displayed. But suppose it is clear early on that tof should started with a smaller value (it started at -224 and increasing negative, and it is clear the best value is at a smaller negative number, such as -221). Do the following:

```
aa                              ;stops the acquisition
satfrq=0                        ;removes problems with diagonal array
array(‘tof’,24,-214,-0.8)       ;re-array’s tof to a better set; use same step as before
arraytof                        ;sets satfrq=tof and re-diagonalizes the array
au                              ;restarts the acquisition
```

The **au** command is required. **au** is identical to **go** except that it will run **wexp** at the completion of the acquisition. Note that the macro **BPautowatergateproc3** sets up a **tof** array, sets **wexp=’BPautowatergateproc3a’**, and then issues an **au**. So an experiment is started, and at the end of the acquisition, **BPautowatergateproc3a** is run. This macro first analyzes the data, changes the value of **tof** appropriately (reducing the array steps for **tof**), sets a new **wexp** and starts (using **au**) the new experiment.

In this way, experiments are chained serially.

III. Referencing the Indirect ^{13}C Axis in BioPack’s gChsqc Experiments:

The **gChsqc** sequence providing in BioPack is huge, with many variations (flags) to switch between band-selection modes, as well as a number of other (esp. relaxation) methods. In many ways, this internal and mostly hidden complexity eases use for researchers. But referencing of the indirect ^{13}C axis is nontrivially complicated. The details are clearly pointed out in the sequence and man page documentation. Even so, the changes are not easy to understand, or correct in the resulting data. The following should guide in gaining understanding, and provides a recipe for applying a correct reference:

1. **<u>What is the problem?</u>** The **gChsqc** sequence (code found at /vnmr/psglib/gChsqc.c) changes the decoupler (^{13}C) frequency independent of any single experiment parameter. In particular, the following code changes are made:

   ```
   if allC=’y’ then        dofa = dof + 35ppm
   if aliph=’y’ then       dofa = dof
   if alphaC=’y’  then     dofa = dof + 21ppm
   if arom=’y’ then        dofa = dof + 90ppm
   ```

   followed later by a statement:

   ```
   decoffset(dofa);
   ```

Since the experiment is run with the decoupler set to **dofa**, and this value is not reflected anywhere in the parameters (but only indirectly by the band-selection flags shown above), absolute referencing by macros such as **decref** will not work properly [see the “Xi Referencing” guide for more details about **xref** and **decref**].

decref uses **sfrq** the ^{1}H (F2) to calculate the absolute frequency at 0 ppm in the ^{1}H (F2) dimension, multiplies by the Xi value for ^{13}C to obtain the absolute value for 0 ppm on that (F1) axis. It then sets 0 ppm there based on the **dfrq** value. But **dfrq** depends on **dof**; the changes made above with **dofa** are not taken into account by **decref**. Thus, the following steps (or equivalent) should be made to correct this situation:

2. **Step-by-step correction:**

a) Find the value of **dof** corresponding to 35 ppm:

i) Acquire a ^{1}H spectrum, and properly reference it.

ii) In another exp#, acquire a nt=1 ^{13}C spectrum. No peaks need to be observed. Reference this spectrum using **xref**.

iii) Set **cr=35p↵** and use **movetof↵**

The value of **tof** in this experiment is the correct value to set **dof** in gChsqc.

b) Acquired the gChsqc using correct values of **dof**, and band-selective flags.

c) Process the data with **wft2da**, phase, etc.

d) Make sure the ^{1}H (F2) axis is referenced correctly.

e) Run **decref↵** (note that this references to DSS).

f) Set **cr1=0↵**

g) Correct the reference using the offsets, as described above:

if **allC**=’y’ then	**rl1(35d)↵**
if **aliph**=’y’ then	no changed needed
if **alphaC**=’y’ then	**rl1(21d)↵**
if **arom**=’y’ then	**rl1(90d)↵**