Journal Name

Dynamic Article Links ►

Cite this: DOI: 10.1039/c0xx00000x

www.rsc.org/xxxxxx

ARTICLE TYPE

Ultra-high dispersion NMR reveals new levels of detail

Juan A. Aguilar,[*a] Peter Kiraly,[b] Ralph Adams,[b] Mickaëlle Bonneau,[a] Elizabeth J. Grayson,[a] Mathias Nilsson,[b] Alan M. Kenwright[a] and Gareth A. Morris[b]

Received (in XXX, XXX) Xth XXXXXXXXX 20XX, Accepted Xth XXXXXXXXX 20XX

DOI: 10.1039/b000000x

Pure shift NMR techniques can provide exquisite resolution, enabling chemists to use ^{1}H NMR to analyse samples that would otherwise require unrealistically high magnetic fields. Although powerful, such techniques remain little used, in part because of the false perception that they are unsuitable for daily use. The present paper shows that this need not be the case and that high quality results can be produced routinely even for dilute and highly complex samples.

Every day very large numbers of ^{1}H NMR spectra are acquired both in academia and industry: ^{1}H NMR is the main workhorse of chemical structure determination. One common problem with ^{1}H NMR spectra is that signals overlap. The low dispersion of proton signals and the extra space needed to accommodate signal multiplicity often conspire to make such spectra hard to interpret. One remedy is often two-dimensional NMR. Experiments such as ^{1}H-^{13}C HSQC[1] that exploit the large dispersion of the carbon dimension provide the best resolving power for most molecules. Unfortunately, many samples remain difficult to analyse even with these methods. Typical problems that are frequently encountered are those in which structural similarity between different parts of a molecule is present. Carbohydrates, complexes, lipids, aromatic systems and oligomers are typical examples. Naturally, the presence of multiple species makes the problem even worse. Some cases lead to such difficulties that analysis is often not even attempted.

There is little that can be done to deal with the poor chemical shift range of protons, but, fortunately, signal dispersion can be improved by collapsing proton multiplets into singlets using "pure shift" methods.[2,3] This approach can be highly efficient: to achieve similar resolution by the usual means of increasing the magnetic field would require fields that are way beyond current engineering capabilities.[4,5] However, in spite of the great performance advantage of pure shift methods, their potential remains largely untapped; they are as yet relatively little used. They are two reasons for this. The first is that there is a mismatch between what a chemist considers a typical sample and the samples that have been used to demonstrate pure shift methods. Most samples run by NMR services catering for chemists are dilute, impure (only end products tend to be pure), they often contain unknowns, and are sometimes unstable. In contrast, samples used for demonstrating the performance of new NMR techniques tend to be concentrated, pure, of known composition, and stable. The second reason is the common assumption that pure shift experiments are complex and time-consuming, and not suitable for routine service use.

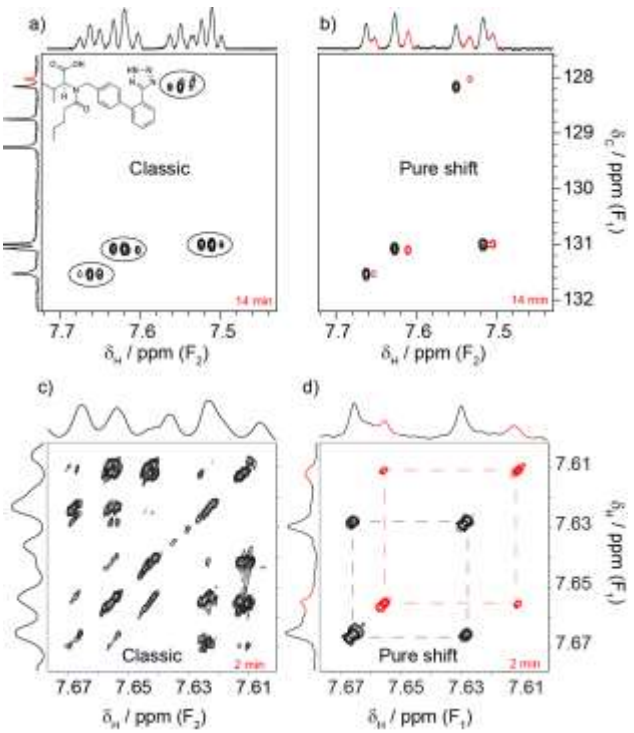


Figure 1. These spectra illustrate how pure shift NMR can reveal a level of detail that conventional experiments lack. The sample is an anti-hypertensive drug, valsartan. There is little indication in the classic HSQC spectrum (a) that the spin system is duplicated. This becomes clear when the proton splittings are removed in the pure shift version (b). The extra spin system is shown in red. The same improvement applies to the COSY ((c) conventional and (d) pure shift) and the ^{1}H (top and side of the COSY) spectra. Many such cases are likely to go unnoticed in everyday investigations.

The purpose of this paper is to show, as for example in Figures 1 to 3 below, that this assumption is both incorrect and unfortunate: many problems that could be solved using pure shift methods go unnoticed. It is precisely in routine NMR facility use that such methods can make a major impact on research projects that require frequent NMR analysis.[6] Furthermore, it is not true that all pure shift experiments require excessively long times. The sensitivity of current hardware means that most samples can be run within modest time allocations. As will be shown here, impressive results can be produced routinely even with samples

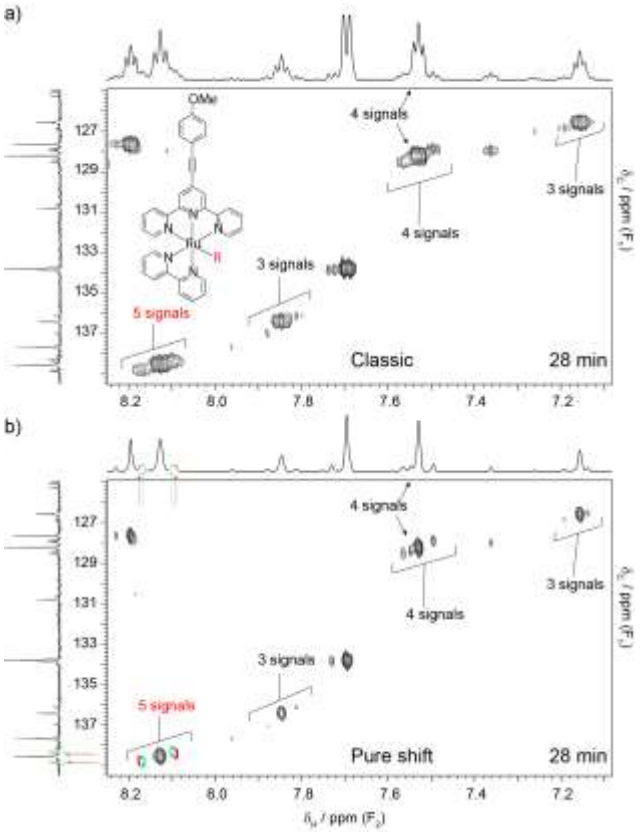


Figure 2. Conventional (a) and pure shift (b) HSQC spectra of a metallo-photo-catalyst whose structure is depicted in (a). The number and nature of species present in solution was of interest because the activity of the catalyst depends on the replacement of the labile group (-R, originally a triflate), with weak chelators. It was also of interest because degradation products compromise the performance of the catalyst. Replacement of the labile group is characterised by patterns of peaks that shadow one another, due to small chemical shift changes. On the other hand, degradation products produce signals that deviate from such patterns. The shadowing pattern can be seen in both experiments, but the pure shift version (b) reveals more species that its conventional counterpart (a). Each spectrum took 28 min to acquire, while the carbon spectrum took 3 h. The total catalyst concentration was 7.5 mM, made up of multiple species.

of uncertain composition, often dilute, sometimes unstable, and always complex. The target audience is that of researchers dealing with small- to medium-sized molecules.

The examples presented here are all drawn from an NMR service that caters for typical synthetic chemists. All the samples formed part of real investigations, and had to be analysed within the time-constrained schedule of such a service. In all cases, sample composition turned out to be more complex than initially anticipated, and in some cases, there was no indication that pure shift methods were necessary. The three illustrative experiments used are a one-dimensional pure shift ^{1}H, a pure shift COSY and a modified real-time pure shift ^{1}H-^{13}C HSQC experiment. One-dimensional spectra were produced using a Zangger-Sterk[5] (ZS) pulse sequence, although the recently reported PSYCHE method[6] could have been used to advantage, as it has higher sensitivity and does not require a compromise to be made between sensitivity and decoupling bandwidth. The COSY experiment used a single quantum constant-time sequence instead of the previously reported multiple-quantum filtered experiment,[8] as it is more sensitive. PSYCHE-TOCSY[9] could also been used following similar arguments to the 1D experiment. Both constant-time COSY and PSYCHE-TOCSY are less sensitive than their traditional counterparts, but this is normally not a problem for the target audience provided that large constant times in the first case and long pulses in the second are avoided. Combining these experiments with covariance processing[9] further simplifies spectra, as in the examples presented, but this is optional. For HSQC, a variant of a previous version[11] was used differing only in that adiabatic pulses were used to improve the off-resonance performance of the carbon pulses. All pulse sequences can be found in the supplementary information (SI).

Valsartan, an angiotensin II receptor antagonist, is a good example of why it is helpful to acquire pure shift spectra routinely. The drug was being investigated to determine whether its conformations in solution correlate with the existence of polymorphs in solid formulations. Part of the investigation focused on the spin system shown in Figure 1a. The spin system is duplicated due to the presence of different conformers, but this duplication went unnoticed even when a combination of conventional experiments was used (^{1}H, ^{1}H-^{1}H COSY, ^{13}C{^{1}H} and ^{1}H-^{13}C HSQC). In contrast, such duplication became clear when pure shift variants were used (same figure). 20 min was all that was needed to break the problem using the pure shift experiments.

Time restrictions are a fact of life for NMR facilities, and users often struggle to solve the problem at hand under typical time constraints. The problem is aggravated when samples are suspected of being unstable, as in the next example. Here, the object of the investigation was the photo-catalyst of Figure 2a. There was an interest in determining how many different complexes existed in solution as the efficiency of the catalyst depends on the nature and stability of such complexes. All the complexes were expected to have the same structure, apart from one labile group (–R). This group was initially a triflate, but was expected to be replaced with other species present in solution. This type of problem is very difficult because all the species are very similar and produce signals that are very close to one another. The utility of the catalyst also depended on its stability, so the need to identify possible degradation products further complicated matters. Cases such as the present one, can be analysed using ^{1}H-^{13}C HSQC experiments, because signals from similar complexes produce patterns only slightly displaced from one another while degradation products tend to produce at least some signals with significant deviations. The problem is that conventional ^{1}H-^{13}C HSQC can only be successful if the carbon dimension is very well digitised, which is very time-consuming. In this particular case, adequate digitisation would have required 4 to 5 h of spectrometer time. This was not available because little time was left after three hours of spectrometer time were already consumed acquiring a ^{13}C{^{1}H} that did not provide much insight. In contrast, a pure shift HSQC spectrum provided the necessary resolving power within 28 min, showing that at least

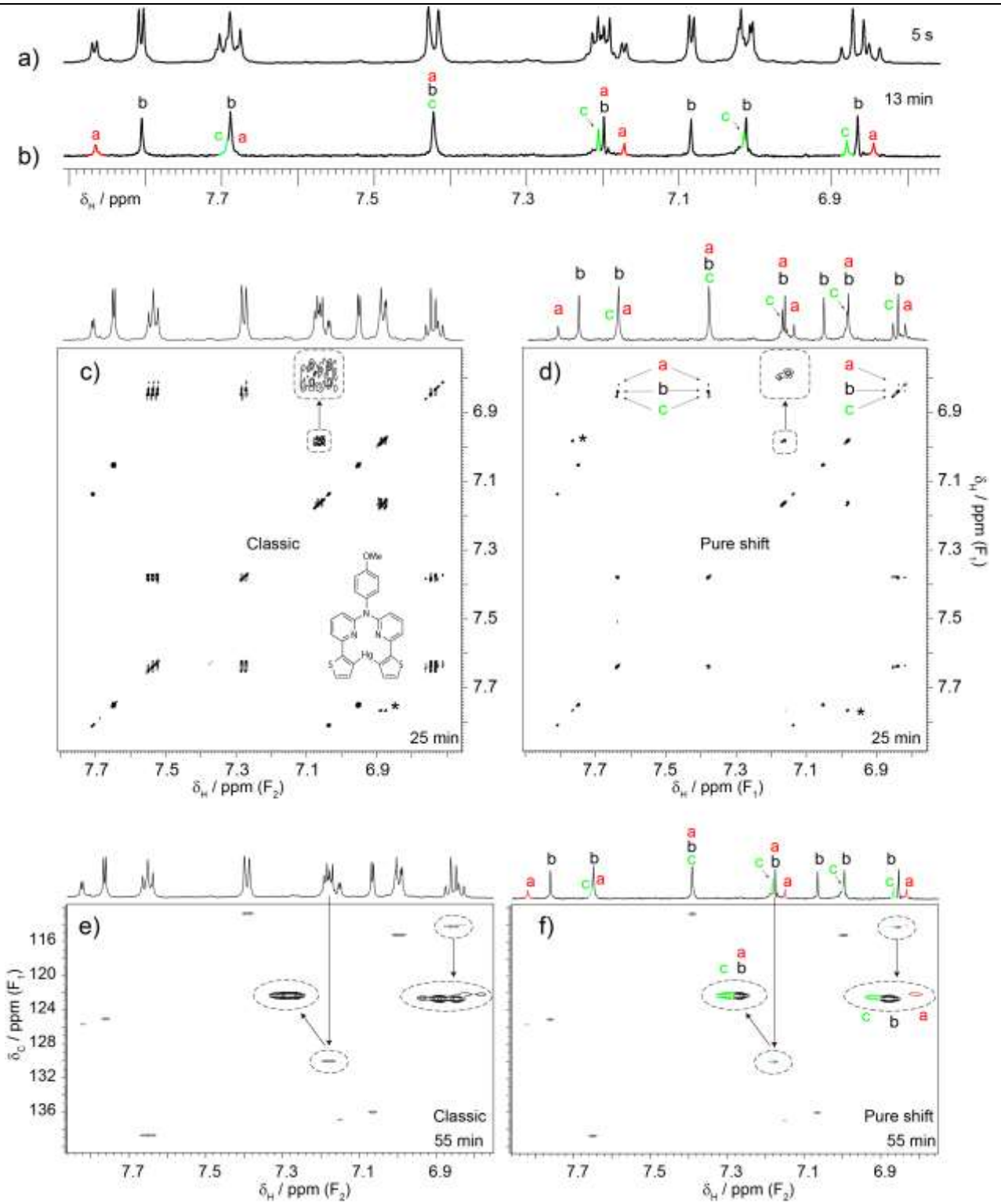


Figure 3. An example of how the proposed combination of experiments was used to solve a problem involving a dilute sample containing unexpected components. A researcher was analysing the ^{1}H, COSY and HSQC spectra of a sample containing the expected mercury complex shown in (c). The researcher was only suspicious of the 7.85 ppm signal but a colleague, suspecting that this was not the only problem, acquired a pure shift ^{1}H. It revealed the presence of more signals than could be attributed to the expected product. Some signals were doubled or even tripled, indicating that the sample contained at least three species. This led to the acquisition of pure shift COSY and HSQC experiments. These revealed that other signals that appear as singlets in the one-dimensional pure shift spectrum were mixtures of overlapping signals. Notice that their correlation patterns are quite similar even in the HSQC spectra, thus indicating structural relatedness. An extended analysis indicated the presence of the uncoordinated ligand and two complexes. The free ligand was actually present but in exchange with at least one of the complexes. It was this dynamic process that prevented the researcher from detecting the presence the free ligand. Note that the combination of pure shift ^{1}H, COSY and HSQC made the analysis possible. The pure shift experiments took a total of 1.5 h of spectrometer time although only one should have sufficed. *Notice how a little more than one hour of spectrometer time made an important impact on this particular investigation. The total concentration was 1.6 mM but three species are present.* The cross-peak labelled with an asterisk is a folded peak.

five very similar species were present. Notice how major peaks are shadowed by minor ones, producing the repetitive pattern noted earlier (see Figure 2b.) It is remarkable that this was accomplished in spite of the short time available and of the low concentration of the sample, 7.5 mM shared among the different species. This was possible because while the resolving power of the pure shift HSQC experiment increases only linearly with experiment duration in the carbon domain, the large resolution improvement afforded by pure shift acquisition comes at no cost in experiment time. A variety of similar examples from our NMR service can be seen in the supplementary information section.

Although in the previous case a single experiment was enough, it is generally better to use the three proposed experiments together. A case that typifies this situation is that of Figure 3. Here another researcher was making a biomedical imaging probe[12] that required formation of the mercury complex shown. The unexpected form of the signal at 7.85 ppm led to the measurement a 1D pure shift ^{1}H spectrum. Unexpectedly, this revealed that some signals were doubled or tripled, indicating that the sample contained at least three species (see for example the 6.8-6.9 ppm cluster of peaks). The task was then to determine the identities of these species. It was necessary to correlate and group those extra signals, but this was not possible using conventional COSY. Double pure shift COSY revealed that some signals that appear as singlets in the pure shift ^{1}H spectrum actually originate from mixtures of signals that show similar correlation patterns. This becomes clear when all three complementary pure shift experiments are used in conjunction. Further analysis indicated the presence of the free ligand and two complexes, meaning that the sample was unsuitable for its intended use. However, the experiments also indicated that the problem could be solved by repeating the reaction with a higher mercury to ligand ratio. The reaction was attempted and indeed a single product was obtained. The take-home message here is not that pure shift experiments revealed some hidden signals, but that they allowed analysis at a level of detail that made it possible to identify an unsuspected problem with the synthesis, and subsequently optimise the synthesis to avoid the problem.. Again, this was done with a sample that contained less than 1 mg mL^{-1} of material (i.e. sub-mM concnetration) using a conventional (not cryogenically-cooled) probe. All the experiments took the same time to run as their classic counterparts, with the exception of the pure shift ^{1}H 1D spectrum, which took 13 min. The pure shift HSQC was left to run for 55 min., but 30 would have been sufficient, and the constant-time COSY took 25 min. To put these experiments in context, a standard ^{13}C{^{1}H} 1D spectrum was acquired. Chemists typically rely on such experiments to provide resolving power, yet after 4 h the resulting spectrum (included in the supplementary information section) was barely usable.

Conclusions

In all the cases presented, pure shift methods revealed a level of detail that was unobtainable using conventional NMR experiments. In all cases, the pure shift experiments fitted well within the typical schedule of the NMR service, and indeed in some cases required less time than conventional experiments. The potential of pure shift experiments as enablers of better and more efficient investigations is only going to grow in the near future, and there is much to be gained by early adoption. Used in conjunction with emerging time-saving approaches such as non-uniform sampling[13] and with high sensitivity probes,[14] such experiments are poised to deliver a minor revolution in NMR.

The authors are indebted to the Durham University NMR service, R. Goodwill, Prof P. Low, Prof P.G. Steel and Prof J.A.G. Williams.

Notes and references

[a] *Department of Chemistry. Durham University. South Road, Durham, DH1 3LE (UK). E-Mail: j.a.aguilar@durham.ac.uk*
E-mail: j.a.aguilar@durham.ac.uk
[b] *School of Chemistry. University of Manchester. Oxford Road, Manchester M13 9PL (UK).*

† Electronic Supplementary Information (ESI) available: 1- The ^{13}C{^{1}H} acquired to serve as a reference for the example of Figure 3. 2- Pure shift spectra of selected samples submitted to the Durham NMR service. 3- Pulse sequences and macros used in the present publication
- The real-time HSQC sequence for Agilent spectrometers.
- The constant-time COSY pulse sequence for Agilent spectrometers.
- The 1D Zangger-Sterk pulse sequence for Agilent spectrometers.
- The reconstruction macro necessary to assemble the raw data produced by the Zangger-Sterk pulse seq]. See DOI: 10.1039/b000000x/

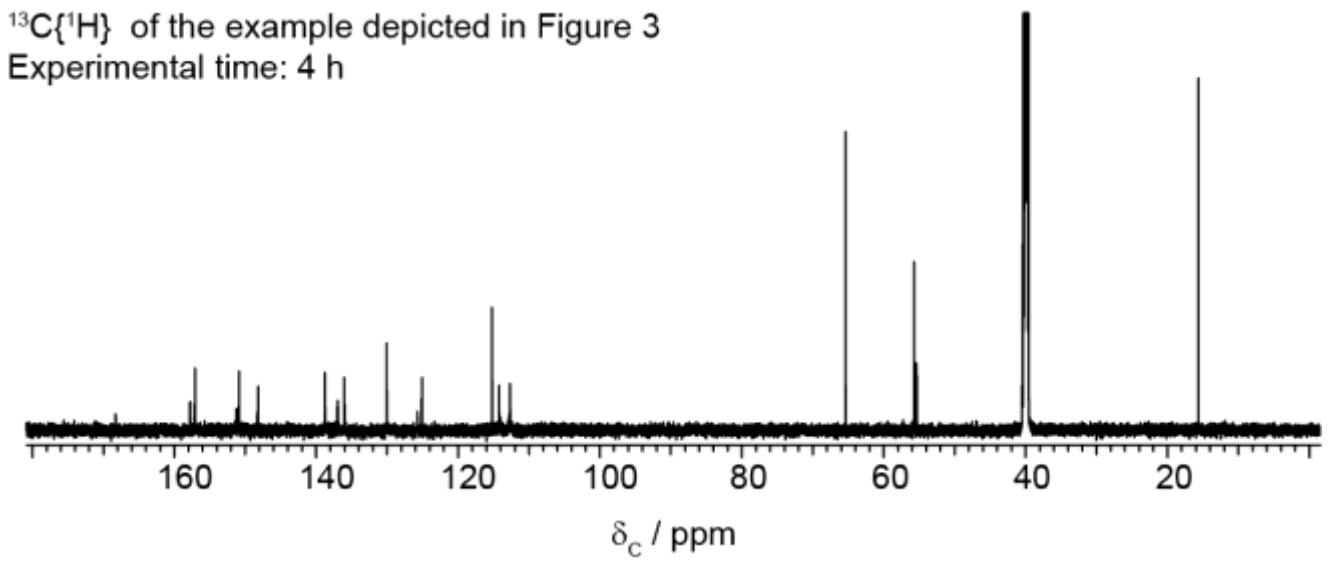
[1] G. Bodenhausen and D. J. Ruben, *Chem. Phys. Lett.*, 1980, 69, 185.
[2] N. H. Meyer and K. Zangger, *ChemPhysChem*., 2014, **15**, 49.
[3] R. W. Adams, *eMagRes*, 2014, **3**, 295. DOI: 10.1002/9780470034590.emrstm1362.
[4] K. Zangger and H. Sterk, *J. Magn. Reson.* 1997, **124**, 486
[5] J. A. Aguilar, S. Faulkner, M. Nilsson and G. A. Morris, *Angew. Chem. Int. Ed.*, 2010, **122**, 3901.
[6] M. Foroozandeh, R. W. Adams, N. J. Meharry, D. Jeannerat, M. Nilsson, G. A. Morris, *Angew. Chem. Int. Ed.*, 2014, **53**, 6990.
[7] J. A. Aguilar, J. Cassani, M. Delbianco, R. W. Adams, M. Nilsson and G. A. Morris, *Chem. Eur. J.*, 2015, accepted. DOI: 10.1002/chem.201406283
[8] J. A. Aguilar, A. A. Colbourne, J. Cassani, M. Nilsson, and G. A. Morris, *Angew. Chem. Int. Ed.* 2012, **51**, 6460.
[9] M. Foroozandeh, R. W. Adams, M. Nilsson and Gareth A. Morris, *J. Am. Chem. Soc.*, 2014, **136**, 11867.
[10] R. Brüschweiler, F. Zhang, *J. Chem. Phys.*, 2004, **120**, 5253.
[11] L. Paudel, R. W. Adams, P. Király, J. A. Aguilar, M. Foroozandeh, M. J. Cliff, M. Nilsson, P. Sándor, J. P. Waltho, G. A. Morris, *Angew. Chem. Int. Ed.*, 2013, **52**, 11616.
[12] W. A. Tarran, G. R. Freeman, L. Murphy, A. M. Benham, R. Kataky and J. A. G. Williams, *Inorg. Chem.*, 2014, **53**, 5738 .
[13] M. Mobli and J. C. Hoch, Spraul, *Prog. Nucl. Magn. Reson. Spectrosc.* 2014, **83**, 21.
[14] H. Kovacs, D. Moskau and M. Spraul, *Prog. Nucl. Magn. Reson. Spectrosc*. 2005, **46**, 131.

Ultra-high dispersion NMR reveals new levels of detail.

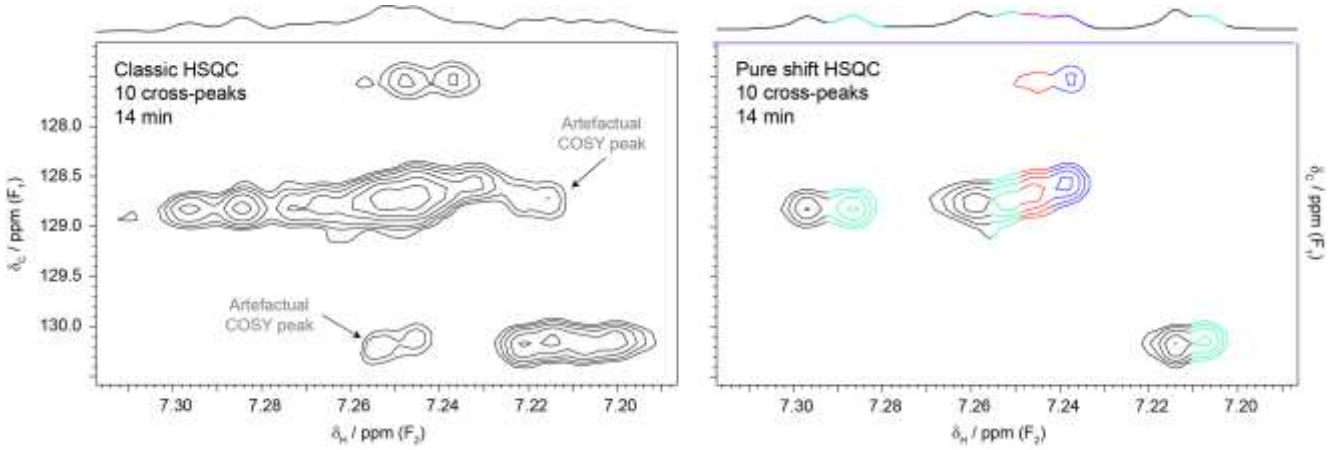
Juan A. Aguilar,[*a] Peter Kiraly,[b] Ralph Adams,[b] Mickaëlle Bonneau,[a] Elizabeth J. Grayson,[a] Mathias Nilsson,[b] Alan M. Kenwright[a] and Gareth A. Morris[b]

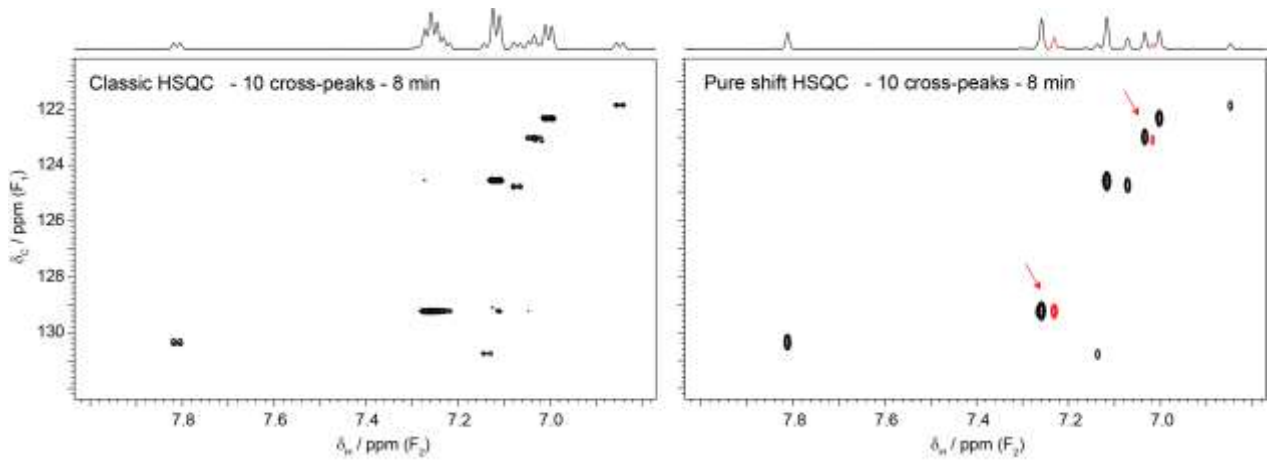
Supplementary information content

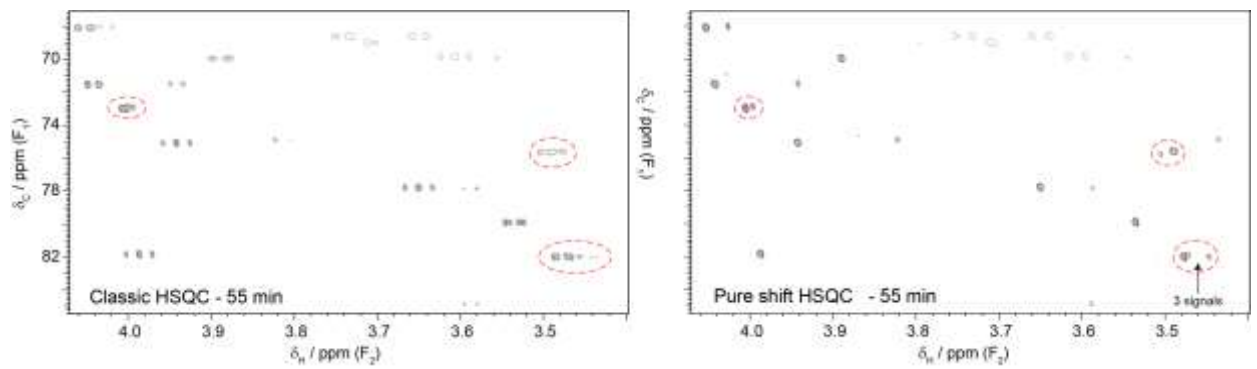
1- The $^{13}C\{^{1}H\}$ acquired to serve as a reference for the example of Figure 3.
2- Pure shift spectra of selected samples submitted to the Durham NMR service.
3- Pulse sequences and macros used in the present publication
- The real-time HSQC sequence for Agilent spectrometers.
- The constant-time COSY pulse sequence for Agilent spectrometers.
- The 1D Zangger-Sterk pulse sequence for Agilent spectrometers.
- The reconstruction macro necessary to assemble the raw data produced by the Zangger-Sterk pulse sequence.

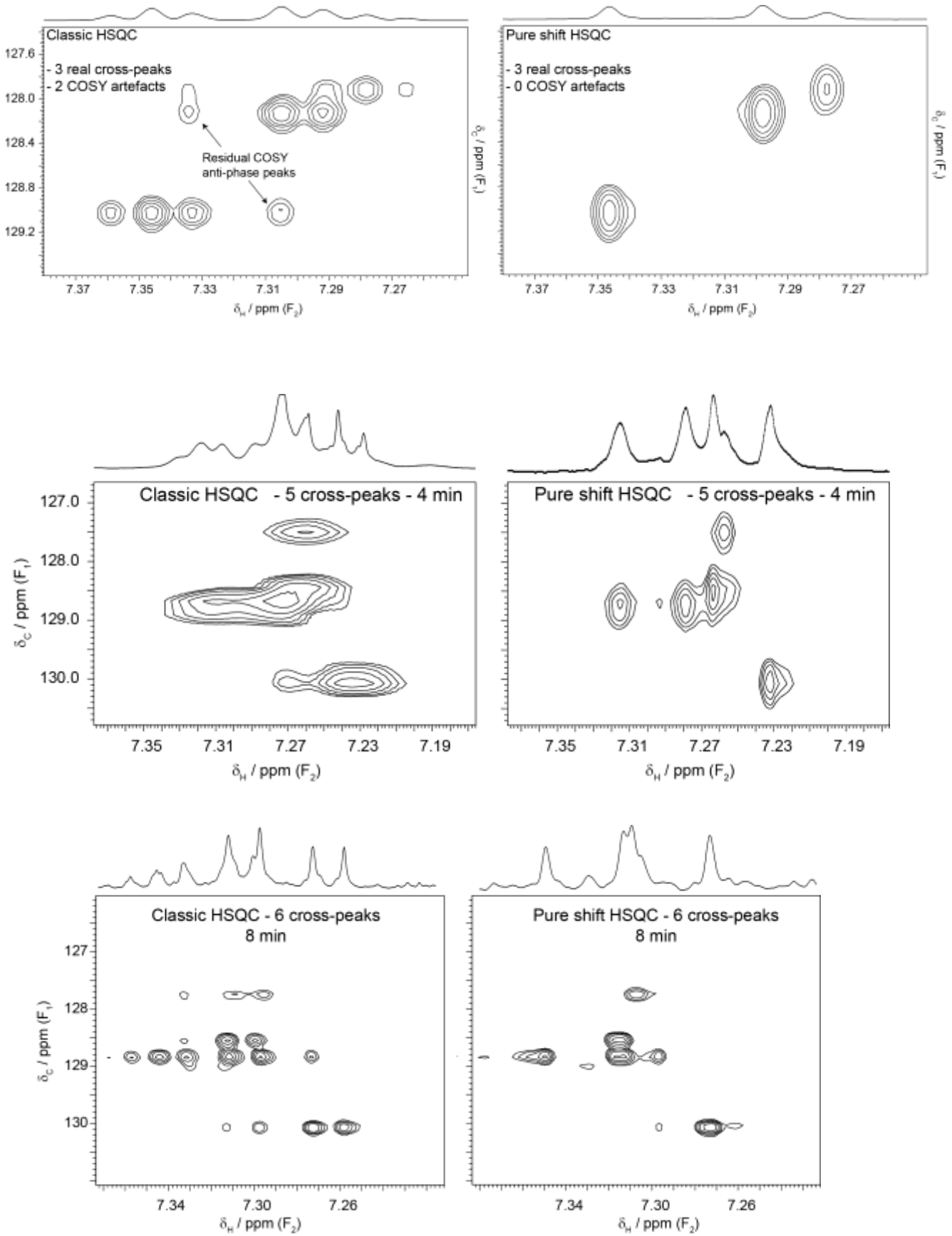


Pure shift spectra of selected samples submitted to the Durham NMR service.









```
/* The real-time HSQC pulse sequence for Agilent spectrometers STARTS here. Delete this
line and save the file as /psglib/rtpsHSQC.c */

*------------------------------------------------------------------------
-------------------------------------------
  This code is provided as a record of the pulse programmes used to obtain the spectra
reported in this publication.  There is no warranty (implied or explicit) that it is optimal or bug-
free.  Anyone using this code does so at their own risk.

Developed by the Manchester NMR Group (University of Manchester) and
modified by J. A. Aguilar (Durham University) to use only proton squares
pulses during the real time acquisition when BIPmode='b'

University of Manchester
School of Chemistry
University of Manchester
United Kingdom
May 2013
This is an experimental pulse sequence. Use it accordingly.

2013-07-04 option for adiabatic ('n'/'y') HSQC was added; adjust f1coef
accordingly

------------------------------------------------------------------------
User's Guide for experimental setup:
------------------------------------------------------------------------
1. BIRD = 'n' selects conventional gHSQC
------------------------------------------------------------------------
2. BIRD = 'y' selects real-time pure shift gHSQC (gHSQC-BIRD)
Three options within BIRD are:
BIRDmode ='h' selects hard 13C inversion pulse during BIRD // large off-
resonance effect, not recommended
BIRDmode ='b' selects BIP 13C inversion pulse during BIRD
BIRDmode ='w' selects a pair of wurst adiabatic 13C inversion pulses during
BIRD
For all np should be integer submultiple of npoints
Users control chunking time using npoints so that np/npoints is an integer
Note:
chunk_time=npoints/(2*sw)=at/cycles
cycles=np/npoints, an integer
at=np/(2*sw)=cycles*npoints/(2*sw)=cycles*chunk_time

JAA: mult=0,1,2 As in the regular experiment.
------------------------------------------------------------------------
-----------------------------------------------------*/

#include <standard.h>
//#include <chempack.h>

/*-----------------------------------------
Phase tables for Varian gHSQC
```

```
-------------------------------------------*/

static int ph1[4] = {1,1,3,3}, //v1 - proton 90 at the end of first inept
ph2[2] = {0,2}, //v2 - X 90 at the end of first inept
ph3[8] = {0,0,0,0,2,2,2,2}, //v3 - proton 90 in 2nd inept
ph4[16] = {0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2}, //v4 - X 90 in 2nd inept
ph5[16] = {1,3,3,1,3,1,1,3,3,1,1,3,1,3,3,1}; //oph
/*-------------------------------------------
Phase tables for rtgHSQC-BIRD
-------------------------------------------*/
static int ph11[8] = {1,1,1,1,3,3,3,3}, //v1
ph12[2] = {0,2}, //v2
ph13[16] = {0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2}, //v3
ph14[32] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2}, //v4
ph15[32] =
{1,3,1,3,3,1,3,1,3,1,3,1,1,3,1,3,3,1,3,1,1,3,1,3,1,3,1,3,3,1,3,1}; //oph
static int ph17[4] = {0,0,1,1}, //v7 - 1st 90 of bird and the hard 180
ph18[4] = {1,1,2,2}, //v8 - simpulse 180 of bird
ph19[4] = {2,2,3,3}; //v9 - 2nd 90 of bird

pulsesequence()
{
/*-----------------------------------------------------
DECLARE AND LOAD VARIABLES
-----------------------------------------------------------*/
//HSQC part

double evolcorr=2.0*pw+4.0e-6,
tau = 1.0/(4.0*(getval("j1xh"))),
taug=2.0*tau,
mult = getval("mult");

int phase1 = (int)(getval("phase")+0.5),
ZZgsign=1.0,
icosel;

//BIRD
double
rof3=getval("rof3"), //delay for receiver off - can be zero if ddrpm='r'
tauA=getval("tauA"), //compensation for tauB and tauC
tauB=getval("tauB"), //effect of rof2
tauC=getval("tauC"), //effect of alfa
tBal=getval("tBal"), //supports inova console if ~1/(fb*1.3)
pwr_XBIP = getval("pwr_XBIP"),
pwr_HBIP = getval("pwr_HBIP"),
pw_XBIP = getval("pw_XBIP"),
pw_HBIP = getval("pw_HBIP"),
npoints=getval("npoints"), // npoints should be an integer multiple of np
cycles=np/npoints;
cycles = (double)((int)((cycles)));
initval(cycles,v20);
```



```
char shp_HBIP[MAXSTR],
shp_XBIP[MAXSTR];
getstr("shp_HBIP",shp_HBIP);
getstr("shp_XBIP",shp_XBIP);

//extensions for AD
double pwx180 = getval("pwx180"),
pwxlvl180 = getval("pwxlvl180"),
pwx180r = getval("pwx180r"),
pwxlvl180r = getval("pwxlvl180r");

char pwx180ad[MAXSTR],
pwx180adR[MAXSTR],
pwx180ref[MAXSTR];
getstr("pwx180ad", pwx180ad);
getstr("pwx180adR", pwx180adR);
getstr("pwx180ref", pwx180ref);

//gradients
double gtE = getval("gtE"), //HSQC encoding
gzlvlE = getval("gzlvlE"),
gstab = getval("gstab"),
gtD = getval("gtD"), //HSQC decoding
gzlvlD = getval("gzlvlD"),
hsglvl = getval("hsglvl"),
hsgt = getval("hsgt"),
hsgstab = getval("hsgstab");

char BIRD[MAXSTR], // Flag to choose gHSQC/rtgHSQC-BIRD ('n'/'y')
BIRDmode[MAXSTR], //Flag to choose hard/bip/wurst2i ('h'/'b'/'w')13C
inversion pulse within BIRD
adiabatic[MAXSTR]; //Flag to use adiabatic refocusing on X-channel
('n'/'y')
getstr("BIRD",BIRD);
getstr("BIRDmode",BIRDmode);
getstr("adiabatic",adiabatic);

char sspul[MAXSTR],
PFGflg[MAXSTR];
getstr("sspul",sspul);
getstr("PFGflg",PFGflg);

//evolcorr and mult declarations
if (adiabatic[0]=='n')
{
evolcorr = 2*pw+4.0e-6;
if (mult > 0.5)
taug = 2*tau;
else
taug = gtE + gstab + 2*GRADIENT_DELAY;
ZZgsign=-1;
if (mult == 2) ZZgsign=1;
icosel = 1;
```

```
}
//AD
if (adiabatic[0]=='y')
{
evolcorr = (4*pwx/PI)+2*pw+8.0e-6;
if (mult > 0.5)
taug = 2*tau; // + getval("tauC");
else
taug = gtE + gstab + 2.0 * GRADIENT_DELAY;
ZZgsign=-1;
if (mult == 2) ZZgsign=1;
icosel = 1;
}


//setup the phase cycle
assign(ct,v10);


if (BIRD[0]=='n')
{
//gHSQC phases
settable(t1,4,ph1);
settable(t2,2,ph2);
settable(t3,8,ph3);
settable(t4,16,ph4);
settable(t5,16,ph5);
}
else
{
//rtgHSQC-BIRD phases
settable(t1,8,ph11);
settable(t2,2,ph12);
settable(t3,16,ph13);
settable(t4,32,ph14);
settable(t5,32,ph15);
settable(t7,4,ph17);
settable(t8,4,ph18);
settable(t9,4,ph19);
getelem(t7, v10, v7);
getelem(t8, v10, v8);
getelem(t9, v10, v9);
}

getelem(t1, v10, v1);
getelem(t2, v10, v2);
getelem(t3, v10, v3);
getelem(t4, v10, v4);
getelem(t5, v10, oph);

initval(2.0*(double)(((int)(d2*getval("sw1")+0.5)%2)),v5);
if ((phase1 == 2) || (phase1 == 5))
icosel = -1;
```



```
add(v2,v5,v2);
add(oph,v5,oph);

/* BEGIN PULSE SEQUENCE */
status(A);

if (sspul[A] == 'y')
{
if (PFGflg[A] == 'y')
{
obspower(tpwr);
delay(5.0e-5);
zgradpulse(hsglvl,hsgt);
rgpulse(pw,zero,rof1,rof1);
zgradpulse(hsglvl,hsgt);
}
else
{
obspower(tpwr-12);
delay(5.0e-5);
rgpulse(500*pw,zero,rof1,rof1);
rgpulse(500*pw,one,rof1,rof1);
}
}

obspower(tpwr);
decpower(pwxlvl);
txphase(zero);
decphase(zero);
obsoffset(tof);
decoffset(dof);

delay(d1);
delay(5.0e-5);

status(B);

/****** null flag starts here *****/

if (getflag("nullflg"))
{
rgpulse(0.5*pw,zero,rof1,rof1);
delay(2.0*tau);
if (adiabatic[0]=='y')
{
decpower(pwxlvl180);
decshaped_pulse(pwx180ad, pwx180, zero, rof1, rof1);
rgpulse(2.0*pw,zero,rof1,rof1);
}
else { simpulse(2.0*pw,2.0*pwx,zero,zero,rof1,rof1); }
txphase(two);
delay(2.0*tau);
```

```
if (adiabatic[0]=='y')
{
decshaped_pulse(pwx180adR, pwx180, zero, rof1, rof1);
decpower(pwxlvl);
}
rgpulse(1.5*pw,two,rof1,rof1);
txphase(zero);
zgradpulse(hsglvl,hsgt);
delay(hsgstab);
}

/***************************gHSQC or gHSQC part of pure shift starts here
************************/

rgpulse(pw,zero,0.0,0.0);
delay(tau);
if (adiabatic[0]=='y')
{
decpower(pwxlvl180);
decshaped_pulse(pwx180ad, pwx180, zero, rof1, rof1);
rgpulse(2.0*pw,zero,rof1,rof1);
}
else { simpulse(2.0*pw,2.0*pwx,zero,zero,rof1,rof1); }
txphase(v1);
delay(tau);
if (adiabatic[0]=='y')
{
decshaped_pulse(pwx180adR, pwx180, zero, rof1, rof1);
decpower(pwxlvl);
}
rgpulse(pw,v1,rof1,rof1);

zgradpulse(hsglvl,2.0*hsgt);
decphase(v2);
delay(hsgstab);

decrgpulse(pwx, v2, rof1, 2.0e-6);
txphase(zero);
decphase(zero);

delay(d2/2.0); // First half of t1 evolution
rgpulse(2.0*pw,zero,2.0e-6,2.0e-6);
delay(d2/2.0); // Second half of t1 evolution

/**/

if (adiabatic[0]=='y')
{
delay(taug - POWER_DELAY);
if (mult > 0.5)
{
decpower(pwxlvl180r);
decshaped_pulse(pwx180ref, pwx180r, zero, rof1, rof1);
```



```
rgpulse(mult * pw, zero, rof1, rof1);
delay(taug - mult * pw - 2.0*rof1 + POWER_DELAY - gtE - gstab - 2.0 *
GRADIENT_DELAY+evolcorr);
zgradpulse(gzlvlE,gtE);
delay(gstab);
decshaped_pulse(pwx180ref, pwx180r, zero, rof1, rof1);
}
else
{
decpower(pwxlvl180);
decshaped_pulse(pwx180ad, pwx180, zero, rof1, rof1);
delay(taug + POWER_DELAY - gtE - gstab - 2.0 * GRADIENT_DELAY+evolcorr);
zgradpulse(gzlvlE,gtE);
delay(gstab);
decshaped_pulse(pwx180ad, pwx180, zero, rof1, rof1);
}
decpower(pwxlvl);
}

if (adiabatic[0]=='n')
{
zgradpulse(gzlvlE,gtE);
delay(taug - gtE - 2.0*GRADIENT_DELAY);
simpulse(mult*pw,2.0*pwx,zero,zero,rof1,rof1);
delay(taug + evolcorr);
}

decrgpulse(pwx,v4,2.0e-6,rof1);
zgradpulse(ZZgsign*0.6*hsglvl,1.2*hsgt);
txphase(v3);
delay(hsgstab);
rgpulse(pw,v3,rof1,rof1);

if (adiabatic[0]=='y')
{
decpower(pwxlvl180);
decshaped_pulse(pwx180adR, pwx180, zero, rof1, rof1);
//decpower(dpwr);
}
delay(tau - (2.0*pw/PI) - 2.0*rof1);

if (adiabatic[0]=='y')
{
rgpulse(2.0*pw,zero,rof1, rof1);
decpower(pwxlvl180);
decshaped_pulse(pwx180ad, pwx180, zero, rof1, rof1);
//decpower(dpwr);
}
else { simpulse(2.0*pw,2.0*pwx,zero,zero,rof1, rof1); }

zgradpulse(icosel*gzlvlD,gtD);
decpower(dpwr);
delay(tau - gtD - 2.0*GRADIENT_DELAY - POWER_DELAY);
```

```
/*******************************gHSQC part stops and BIRD Acquisition
starts here*************************/

delay(tBal);
//filter delay (Hoult) for inova; adjust tBal manually for the same effect
//delay(1.0/(getval("fb")*1.3))
if (BIRD[0]=='y')
{
setacqmode(WACQ|NZ); //use this line only for vnmrs console; comment this
out in inova


obsblank();
delay(rof2);
startacq(alfa);
}
/*------------------------------------------------------------------------
-------------------------------------------------
Observe the 1st half chunk
------------------------------------------------------------------------
-------------------------------------------------*/
if (BIRD[0]=='y')
{
status(C);
acquire(npoints/2.0,1.0/sw);
rcvroff();
status(B);
obspower(tpwr);
txphase(v7);

/*------------------------------------------------------------------------
Using hard 13C inversion pulse in BIRD
------------------------------------------------------------------------
*/
if (BIRDmode[0]== 'h')
{
rgpulse(pw,v7,rof1,rof1);
decpower(pwxlvl);
delay(2.0*tau);
simpulse(2.0*pw,2.0*pwx,v8,v8,rof1,rof1);
decpower(dpwr);
delay(2.0*tau);
rgpulse(pw,v9,rof1,rof1);
}

/*------------------------------------------------------------------------
--
Using BIP 13C inversion pulse in BIRD
------------------------------------------------------------------------
-*/
if (BIRDmode[0]== 'b')
{
```



```
rgpulse(pw,v7,rof1,rof1);
obspower(tpwr);
if (pwr_XBIP!=pwxlvl) decpower(pwr_XBIP); else decpower(pwxlvl);
delay(2.0*tau);
simshaped_pulse("",shp_XBIP,pw*2.0,pw_XBIP,v8,v8,rof1,rof1);
decpower(dpwr);
delay(2.0*tau);
rgpulse(pw,v9,rof1,rof1);
}

/*------------------------------------------------------------------------
-
Using a pair of wurst2i adiabatic 13C inversion pulses in BIRD
------------------------------------------------------------------------
-*/
if (BIRDmode[0]== 'w')
{
rgpulse(pw,v7,rof1,rof1);
if (pwr_HBIP!=tpwr) obspower(pwr_HBIP);
if (pwxlvl180!=pwxlvl) decpower(pwxlvl180); else decpower(pwxlvl);
txphase(v8); decphase(v8);
delay(2.0*tau);
decshaped_pulse(pwx180ad, pwx180, v8, rof1, rof1);
shaped_pulse(shp_HBIP,pw_HBIP,v8,rof1,rof1);
if (pwr_HBIP!=tpwr) obspower(tpwr);
txphase(v9);
delay(2.0*tau);
decshaped_pulse(pwx180adR, pwx180, v8, rof1, rof1);
decpower(dpwr);
rgpulse(pw,v9,rof1,rof1);
}
txphase(v7);
delay(tauA);
rgpulse(pw*2.0,v7,rof1,rof1); // hard 180 degree refocusing pulse
obsblank();
delay(tauB);
rcvron(); //this includes rof3
delay(tauC);

decr(v20);

/*------------------------------------------------------------------------
----------------------------------------------------
Loops for more chunks
------------------------------------------------------------------------
-----------------------------------------------------*/

starthardloop(v20);
status(C);
acquire(npoints,1.0/sw);
rcvroff();

status(B);
```

```
obspower(tpwr);
txphase(v7);

/*------------------------------------------------------------------------
Using hard 13C inversion pulse in BIRD
------------------------------------------------------------------------
*/
if (BIRDmode[0]== 'h')
{
rgpulse(pw,v7,rof1,rof1);
decpower(pwxlvl);
delay(2.0*tau);
simpulse(2.0*pw,2.0*pwx,v8,v8,rof1,rof1);
decpower(dpwr);
delay(2.0*tau);
rgpulse(pw,v9,rof1,rof1);
}

/*------------------------------------------------------------------------
--
Using BIP 13C inversion pulse in BIRD
------------------------------------------------------------------------
-*/
if (BIRDmode[0]== 'b')
{
rgpulse(pw,v7,rof1,rof1);
obspower(tpwr);
if (pwr_XBIP!=pwxlvl) decpower(pwr_XBIP); else decpower(pwxlvl);
delay(2.0*tau);
simshaped_pulse("",shp_XBIP,pw*2.0,pw_XBIP,v8,v8,rof1,rof1);

decpower(dpwr);
delay(2.0*tau);
rgpulse(pw,v9,rof1,rof1);
}

/*------------------------------------------------------------------------
-
Using a pair of wurst2i adiabatic 13C inversion pulses in BIRD
------------------------------------------------------------------------
-*/
if (BIRDmode[0]== 'w')
{
rgpulse(pw,v7,rof1,rof1);
if (pwr_HBIP!=tpwr) obspower(pwr_HBIP);
if (pwxlvl180!=pwxlvl) decpower(pwxlvl180); else decpower(pwxlvl);
txphase(v8); decphase(v8);
delay(2.0*tau);
decshaped_pulse(pwx180ad, pwx180, v8, rof1, rof1);
shaped_pulse(shp_HBIP,pw_HBIP,v8,rof1,rof1);
if (pwr_HBIP!=tpwr) obspower(tpwr);
txphase(v9);
delay(2.0*tau);
```



```
decshaped_pulse(pwx180adR, pwx180, v8, rof1, rof1);
decpower(dpwr);
rgpulse(pw,v9,rof1,rof1);
}

txphase(v7);
delay(tauA);
rgpulse(pw*2.0,v7,rof1,rof1); // hard 180 degree refocusing pulse
obsblank();
delay(tauB);
rcvron(); //this includes rof3
delay(tauC);

endhardloop();

/*------------------------------------------------------------------------
-------------------------------------------------
Acquisition of last half chunk
------------------------------------------------------------------------
-------------------------------------------------*/
status(C);
acquire(npoints/2.0,1.0/sw);
rcvroff();
endacq();
incr(v20);
}

/****** BIRD ends here for all *****/


/************************ ACQ for conventional gHSQC
*********************************/
else
{
status(C);
}

}
/********************** PULSE SEQUENCE ENDS
HERE***************************************/
```

/* The rtps-HSQC pulse sequence ENDS here. Delete this line and save the file as /psglib/rtpsHSQC.c */

/* The constant-time COSY pulse sequence for Agilent spectrometers STARTS here. Delete this line and save the file as psglib/COSY_CT.c */

/* This code is provided as a record of the pulse programmes used to obtain the spectra reported in this publication. There is no warranty (implied or explicit) that it is optimal or bug-free. Anyone using this code does so at their own risk. */

```
#ifndef LINT
static char SCCSid[] = "@(#)";
#endif


/* Juan A. Aguilar. Durham. 11-07-2013.
Magnitude mode constant-time COSY-60
Usually provides good results when the bandwidth is reduced (less than 5
ppm). Typically a single transient and 512 increments is enough. Make sure
that the resolution along F1 sufficient to be able to tell the difference
between a singlet and a multiplet. Sensitivity can be improved by degassing
samples, but this if often not necessary.
This is an experimental pulse sequence. Use it accordingly
Paramters:

gzlvl1 : Coherence selection gradient level
gt1 : Gradient time
gstab : Recovery delay
pw : 90 degree pulse length at tpwr
d1 : relaxation delay
cti : Constant time period. Automatically
calculated
*/

#include <standard.h>

static int ph1[4] = {0, 2, 0, 2},
ph2[4] = {0, 0, 0, 0},
ph3[4] = {0, 2, 0, 2};

pulsesequence()
{
double gzlvl1 = getval("gzlvl1"),
gzlvl2 = getval("gzlvl2"),
gt1 = getval("gt1"),
gstab = getval("gstab"),
gstab2 = getval("gstab2"),
cti = getval("cti");
```

```
char sspul[MAXSTR];
getstr("sspul",sspul);

settable(t1,4,ph1);
settable(t2,4,ph2);
settable(t3,4,ph3);

getelem(t1,ct,v1);
getelem(t2,ct,v2);
getelem(t3,ct,oph);

initval(2.0*(double)(((int)(d2*getval("sw1")+0.5)%2)),v10);

cti=2.0*(ni*0.5/sw1+gt1*2.0+gstab*2.0+0.0001);

status(A);
delay(d1);

obspower(tpwr);
obsoffset(tof);
delay(rof1);

status(B);

rgpulse(pw, v1, rof1, rof1);
delay(cti*0.5 - d2*0.5+gstab+gt1);
zgradpulse(gzlvl1,gt1);
delay(gstab);
rgpulse(pw*2.0, zero, rof1, rof1);

zgradpulse(gzlvl1,gt1);
delay(gstab);
delay(cti*0.5 + d2*0.5);
zgradpulse(gzlvl1*0.3,gt1);
delay(gstab);
rgpulse(pw*0.666667, v2, rof1, rof2);
zgradpulse(gzlvl1*0.3,gt1);
delay(gstab);

status(C);
}
```

/* The constant-time COSY pulse sequence for Agilent spectrometers ENDS here. Delete this line and save the file as psglib/COSY_CT.c */

/* The Zangger-Sterk pulse sequence for Agilent spectrometers STARTS here. Delete this line and save the file as /psglib/pureshiftZS.c */

/* This code is provided as a record of the pulse programmes used to obtain the spectra reported in this publication. There is no warranty (implied or explicit). Anyone using this code does so at their own risk. */

```
#ifndef LINT
static char SCCSid[] = "@(#)GMZSr.c 19.1 01/13/06 Copyright (c) 1991-1996
Varian Assoc.,Inc. All Rights Reserved";
#endif
/*
* Varian Assoc.,Inc. All Rights Reserved.
* This software contains proprietary and confidential
* information of Varian Assoc., Inc. and its contributors.
* Use, disclosure and reproduction is prohibited without
* prior consent.
*/

/* GMZSnew - simple Zangger-Sterk pure shift with hard 180 before soft */


#include <standard.h>
#include <Pbox_psg.h>

pulsesequence()
{
double p1lvl,
droppts = getval("droppts"), /* number of dummy points to acquire */
gzlvl1=getval("gzlvl1"), /* CTP selection gradient */
gzlvl2=getval("gzlvl2"), /* slice select gradient */
gzlvl_ss=getval("gzlvl_ss"), /* steady-state crusher gradient */
gtss=getval("gtss"), /* steady-state gradient pulse width */
gt1=getval("gt1"), /* CTP gradient pulse width */
gstab=getval("gstab"), /* gradient stabilisation delay */
selpw=getval("selpw"), /* pulse length for the soft 180 */
selpwr=getval("selpwr"); /* power level for the soft 180 */

char sspul[MAXSTR],
selshape[MAXSTR]; /* pulse file for the soft 180 */

getstr("sspul",sspul);
getstr("selshape",selshape);
```

```
/*Check that gstab > 1/sw */
if (gstab<(droppts)/sw)
{
abort_message("gstab should be greater than droppts/sw\n");
}

/*Check that sw1 is an integer submultiple of sw */
if (fabs((sw/sw1)-(double)((int)((sw/sw1)+0.5)))> 0.01)
{
text_message("WARNING: sw1 should be an integer submultiple of sw\n");
}
/*Check that power deposition in gradient coil is not excessive*/
if ( (gzlvl2*gzlvl2*selpw/(gradstepsz*gradstepsz))> 0.01) /* maximum 1% of
full dissipation */
{
abort_message("Slice select gradient gzlvl2 is dangerously high\n");
}


/* LOAD AND INITIALIZE VARIABLES */
getstr("sspul", sspul);

/* CHECK CONDITIONS */

/* CALCULATE PHASES */
mod4(ct,v1);
mod2(ct,v3);
sub(v3,one,v2);
add(v3,one,v4);
add(v4,one,v5);
mod4(v2,v2);
mod4(v3,v3);
mod4(v4,v4);
hlv(ct,v6);
hlv(v6,v7);
hlv(v7,v7);
mod4(v6,v6);
mod4(v7,v7);
add(v6,v7,oph);
dbl(oph,oph);
add(oph,v1,oph);
mod4(oph,oph); /* v1 + 2(v6+v7) */

/* BEGIN ACTUAL PULSE SEQUENCE CODE */
status(A);
if (sspul[0] == 'y')
{
zgradpulse(gzlvl_ss,gtss);
obspower(tpwr);
rgpulse(1000*1e-6, zero, rof1, 0.0e-6);
rgpulse(1000*1e-6, one, 0.0e-6, rof1);
```

```
zgradpulse(gzlvl_ss,gtss*0.6);
}
obspower(tpwr);
hsdelay(d1);
status(B);

rgpulse(pw,v1,rof1,rof2);

delay(d2/2.0);
delay((0.25/sw1)-gt1-gstab);
zgradpulse(gzlvl1*0.5,gt1);
delay(gstab);
rgpulse(pw*2,v7,rof1,rof2); /* Hard 180*/
delay(0.25/sw1);
delay(gstab);
zgradpulse(-gzlvl1*0.5,gt1); /* second CTP pulse*/
delay(gstab);
obspower(selpwr); /* POWER CHANGE (SOFT) */
rgradient('z',gzlvl2); /* SLICE selection*/
shaped_pulse(selshape,selpw, v6, rof1, rof2); /* SLICE selection - Soft 180
*/
rgradient('z',0.0); /* SLICE selection*/
delay(gstab);
zgradpulse(-gzlvl1,gt1); /* third CTP pulse*/
rcvron();
obsblank();
delay(gstab-droppts/sw);
obspower(tpwr); /* Power level change (HARD) */
delay(d2/2.0);

/* detection */
status(C);
}
```

/* The Zangger-Sterk pulse sequence for Agilent spectrometers END here. Delete this line and save the file as /psglib/pureshiftZS.c */

/* The assembler macro for Zangger-Sterk pulse sequences for Agilent spectrometers STARTS here. Delete this line and save the file as /maclib/pureshift_proc */

```
if (($#>0)) then
write('error','Usage: pureshift_proc; takes no arguments')
abort
endif

jexp:$exp,$expname

cptmp('pureshift')
$nfid=ni
if (lsfid>0) then
$droppts=lsfid+1
else
$droppts=1
ENDIF
exists('droppts','parameter'):$ex
IF $ex>0 then
$droppts=droppts
ENDIF
```

```
exists('sw1','parameter'):$ex
IF $ex<1 then "make sure 2D parameters available"
par2d
ENDIF
exists('nchunk','parameter'):$ex
IF $ex>0 then "backward compatibility"
sw1=sw*2/nchunk
groupcopy('current','processed','acquisition')
ENDIF

$npoint=trunc((sw/sw1)+0.5)
$chunk1=$npoint/2
exists('chunk1','parameter'):$ex
IF $ex>0 then
$chunk1=chunk1
ENDIF
if $chunk1=0 then
$chunk1=$npoint
endif
$tmpfile=userdir+'/'+$expname+'/homodec_writefid'
$tmpfile2=userdir+'/'+$expname+'/homodec_fid'
beepoff

$imag=0.0
$real=0.0

exists($tmpfile,'file'):$ex1
IF $ex1>0 then
shell('rm',$tmpfile)
ENDIF

$i=1
REPEAT
writefid($tmpfile,$i)
lookup('file',$tmpfile)
$k=1
repeat
lookup('read'):$temp "read dummy points"
lookup('read'):$temp "read dummy points"
$k=$k+1
until $k>$droppts

$j=1
IF $i<2 THEN
REPEAT
lookup('read'):$imag[$j]
lookup('read'):$real[$j]
$j=$j+1
UNTIL ($j>($chunk1))
ELSE
REPEAT
lookup('read'):$imag[($i-2)*$npoint+$j+$chunk1]
lookup('read'):$real[($i-2)*$npoint+$j+$chunk1]
$j=$j+1
UNTIL ($j>($npoint))
ENDIF
exists($tmpfile,'file'):$ex1
IF $ex1>0 then
shell('rm',$tmpfile)
ENDIF
$i=$i+1
UNTIL ($i>$nfid)

exists($tmpfile2,'file'):$ex1
IF $ex1>0 then
shell('rm',$tmpfile2)
ENDIF

$i=1
REPEAT
write('file',$tmpfile2,'%d %d',$imag[$i],$real[$i])
```

```
$i=$i+1
UNTIL ($i>(($nfid-1)*$npoint+$chunk1))


rm(curexp+'/acqfil/fid')
shell('sleep 1')
makefid($tmpfile2)


setvalue('np',2.0*($npoint*($nfid-1)+$chunk1))
setvalue('fn',np)
setvalue('at', 0.5*np/sw)
groupcopy('current','processed','acquisition')
exists($tmpfile2,'file'):$ex1
IF $ex1>0 then
shell('rm',$tmpfile2)
ENDIF


exists('nchunk','parameter'):$ex
IF $ex>0 then "backward compatibility"
destroy('nchunk','current')
destroy('nchunk','processed')
ENDIF
lb='n' gf=at/2 lsfid=0
fn=4*np ni=0
groupcopy('current','processed','acquisition')
wft aph full vsadj
```

/* The assembler macro ENDS here. Delete this line */