Electronic Supplementary Material (ESI) for ChemComm.

Electronic Supplementary Information

Improving the detection and quantification of low level impurities using ^{19}F NMR

*Pinelopi Moutzouri, Peter Kiraly, Andrew R. Phillips, Steven R. Coombes, Mathias Nilsson and Gareth A. Morris**

Table of Contents

1. **Pulse sequences and phase cycling**

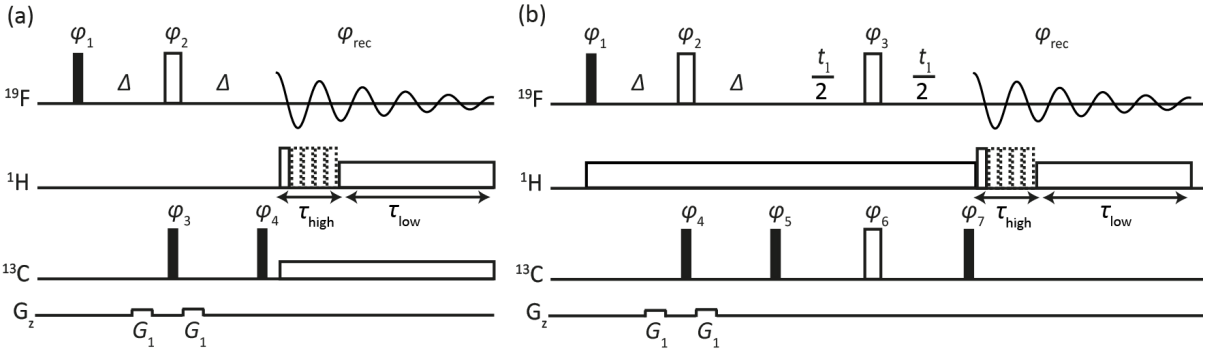

Fig. S1. Pulse sequences for the acquisition of the ^{19}F spectra of (a) Fig. 1b and (b) 1c of the main text.

The experiments reported used the following 16-step phase cycle:

(a)

Phase of Pulse	16-Step Phase Cycle
φ_1	0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3
φ_2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
φ_3	0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2
φ_4	0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2
φ_R	0 0 0 0 3 3 3 3 2 2 2 2 1 1 1 1

Table S1. Phase cycling used for the pulse sequences of Fig. S1a (a) and Fig. S1b (b), where 0,1,2,3 represent rotation about the *x*, *y*, *-x*, *-y* axis, respectively.

(b)

Phase of Pulse	16-Step Phase Cycle
φ_1	0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3
φ_2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
φ_3	0 0 0 0 2 2 2 2 0 0 0 0 2 2 2 2
φ_4	0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2
φ_5	0 0 2 2 0 0 2 2 0 0 2 2 0 0 2 2
φ_6	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
φ_7	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
φ_R	0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3

2. Sample details

The samples used for Fig. 1 of the main text and Fig. S6, contained the compounds listed in Table S2, in DMSO-d_6 supplied by Cambridge Isotope Laboratories Inc.

Compounds	Chemical Shift / ppm	Concentration / mM
(1) Rosuvastatin (supplied by AstraZeneca)	−111.923	37
(2) * BEM (supplied by AstraZeneca)	−111.97	0.121
(3) * DPPO (supplied by AstraZeneca)	−112.707	0.063
(4) Impurity 1a	−111.98	0.104
(5) Impurity 1b (not present in the fresh sample of Fig. S6)	−111.76	0.040

Table S2. Contents, chemical shifts and concentrations of the sample prepared in DMSO-d_6.

* tert-butyl-E-(6-[-[4-(4-fluorophenyl)-6-isopropyl-2-[methyl(methylsulfonyl)amino]pyrimidin-5-yl]-vinyl]-((4R,6S)-2,2-dimethyl[1,3]dioxin-4yl) acetic acid)

* N-(5-((diphenylphosphoryl)methyl)-4-(4-fluorophenyl)-6-isopropylpyrimidin-2-yl)-N-methylmethanesulfonamide

The sample of Fig. 3 of the main text was a 16 mM sample of fluconazole (extracted from Diflucan formulation) in DMSO-d_6.

3. Experiment details

All data were acquired non-spinning on a 500 MHz Varian/Agilent VNMRS spectrometer at a temperature of 25 °C with a 3 mm triple resonance (^{1}H/^{19}F),^{13}C Nalorac probe. This has a (relatively inefficient) double-tuned high band coil and a (highly efficient) inner ^{13}C coil. Unless stated otherwise, the data were acquired with the following conditions.

An RF amplitude of 10 kHz was used for the ^{19}F pulses, 8.6 kHz for the ^{1}H pulses, and 55.5 kHz for the ^{13}C pulses. The 1D spectra of Fig. 1a and 1b were acquired with 32 dummy scans and 4096 transients, using a 5 kHz spectral window, in 13 h

27 min. 8192 complex data points were used, with zero-filling to 65536 complex points. Resolution was enhanced by Lorentz-Gauss conversion with a Gaussian time constant (gf) of 0.53 s and a negative line broadening (lb) of –1.23 Hz. A relaxation delay (d1) of 10 s was used. The 2D experiment from which the 1D proton-decoupled, ^{13}C isotopomer-filtered ^{19}F spectrum of Fig. 1c was extracted was acquired with 32 dummy scans, 256 transients and 16 increments using a 5 kHz spectral window in the direct dimension and a 35 Hz spectral window in the indirect dimension, in 13 h 41 min. 8192 complex data points were used, with zero-filling to 65536 complex points in the direct dimension and to 128 complex points in the indirect dimension. Resolution was enhanced in both dimensions by Lorentz-Gauss conversion with a Gaussian time constant (gf) of 0.53 s in the direct and 0.265 s in the indirect dimension, and a negative line broadening (lb) of –1.08 Hz in the direct and –1.1 Hz in the indirect dimension. A 10 s relaxation delay was used. A pair of pulsed field gradients of 1 ms duration, with an amplitude of 14 G cm^{-1} and a 500 μs gradient recovery delay, was used for CTP selection. The 1D spectrum of Fig. 1c was extracted as a sum projection from F_1 = -1.7 to F_1 = +1.7 Hz of the 2D spectrum of Fig. S5, which is the result of a double Fourier transformation of the 2D time-domain data produced by the sequence of Fig. 2 of the main text in which the imaginary (dispersion mode) part of the data is zeroed after the first Fourier transformation in order to obtain 2D absorption lineshapes. All spectra were acquired with the same bilevel adiabatic proton decoupling during the acquisition period.

The spectra of Fig. 3 were acquired with an RF amplitude of 9.8 kHz for the ^{19}F pulses. The 1D spectrum (Fig. 3a) was acquired with 8192 complex data points and zero-filling to 65536 complex points in 3 h and 51 min. A relaxation delay (d1) of 5 s, 32 dummy scans, 2048 transients and a 5 kHz spectral window were used. Resolution was enhanced by Lorentz-Gauss conversion with a Gaussian time constant (gf) of 0.265 s and a negative line broadening (lb) of -1 Hz. The 2D selective experiments of Fig. 3b and c were acquired on resonance with 8192 complex data points and zero-filling to 65536 complex points in the direct dimension and 128 complex data points in the indirect dimension in 7 h and 57 min. A relaxation delay (d1) of 5 s, 32 dummy scans, 256 transients, 16 increments, a 5 kHz spectral window in the direct dimension and a 35 Hz spectral window in the indirect dimension were used. Resolution was enhanced in both dimensions with a Gaussian time constant (gf) of 0.265 s in the direct and 0.265 s in the indirect dimension, and a negative line broadening (lb) of –1.15 Hz in the direct and –1.4 Hz in the indirect dimension. The 1D spectra were extracted as a sum projection from F_1 = -1.7 to F_1 = +1.7 Hz of the 2D spectra. Both experiments used the following selective pulse, on resonance with the selected signal:

- 180° refocusing selective pulse
  1. Pulse shape: rsnob
  2. Bandwidth: 925 Hz
  3. Duration: 2 ms

4. B_1 max: 1.2 kHz
5. Stepsize: 10 μs

In all proton decoupled experiments the acquisition period was time-shared between ^{19}F detection and ^{1}H decoupling. 60 % of the dwell time was used for decoupling, 37 % for detection, and 3 % for switching the receiver on and off between detection and decoupling. The latter time was determined by the parameters homorof1, homorof2 and homorof3 in the pulse sequence. The architecture used to make this acquisition scheme possible involved setting the preamplifier configuration (preAmpConfig) to "hhn", using the first and second channels for the time-sharing of the ^{19}F detection and ^{1}H decoupling, and the third channel for the application of the ^{13}C pulses. During acquisition an analogue filter was used to reject any ^{1}H radiofrequency during the detection of ^{19}F.

All spectra were acquired with the same bilevel adiabatic proton decoupling during the acquisition period unless stated otherwise. This decoupling scheme used two WURST-2 periods of different durations and amplitudes. In Fig. 2 of the main text the symbol τ_{high} represents the duration for which a high power WURST-2 waveform is applied. This duration is incremented from transient to transient. The symbol τ_{low} represents the duration for which a low power WURST-2 waveform is applied, which is decremented from transient to transient.

- <u>High power WURST-2 decoupling</u>
  1. Wave definition (Pbox): wurst2i
  2. Bandwidth: 4 kHz (bandwidth is a parameter in the pulse sequence)
  3. Duration: 2 ms (the duration is determined by the ratio of the spectral window to npoints, where npoints is a parameter was set equal to 10)
  4. B_1 max: 3.4 kHz
  5. Duty cycle: 0.6
  6. Stepsize: 5 μs (defines the length of a single step in the waveform)
  7. Supercycle: t5[(1)]

- <u>Low power WURST-2 decoupling</u>
  1. Wave definition: WURST2
  2. Bandwidth: 4 kHz
  3. Duration: 4 ms
  4. B_1 max: 1.2 kHz [(2)]
  5. Duty cycle: 0.6
  6. Stepsize: 10 μs

[1] R. Tycko, A. Pines, J. Guckenheimer, *J. Chem. Phys.*, 1985, **83**, 2775-2802
[2] The RF power was increased by an extra 2dB above the minimum power given in the Pbox output, as suggested in S. Cheatham, B. Adams, Ē. Kupče, Magnetic Moments Vol. X.1 – August 1999.

7. Supercycle: m16 (3)

The following low power WURST-2 decoupling wave form is used both during the two Δ delays and during the t_1 evolution period of the experiment of Fig. 1c, in order to suppress any ^{19}F echo modulation caused by the presence of ^{1}H – ^{1}H strong coupling.

- WURST-2 decoupling
  1. Wave definition: WURST2
  2. Bandwidth: 4 kHz
  3. Duration: 4 ms
  4. B_1 max: 0.7 kHz (2)
  5. Stepsize: 10 µs
  6. Supercycle: m16

Multiple-bond carbon couplings were removed in the experiment of Fig. 1b using the following WURST-2 decoupling.

- WURST-2 decoupling
  1. Wave definition: WURST2
  2. Bandwidth: 15 kHz
  3. Duration: 4 ms
  4. B_1 max: 1.5 kHz (2)
  5. Stepsize: 10 µs
  6. Supercycle: t5,m4

[3] M. H. Levitt and R. Freeman, *J. Magn. Reson.*, 1981, **43**, 502-507

4. **Suppressing echo modulation caused by strong coupling**

^{19}F echo modulation caused by the presence of ^{1}H -^{1}H strong coupling can be suppressed by the application of ^{1}H decoupling during the t_1 evolution period.

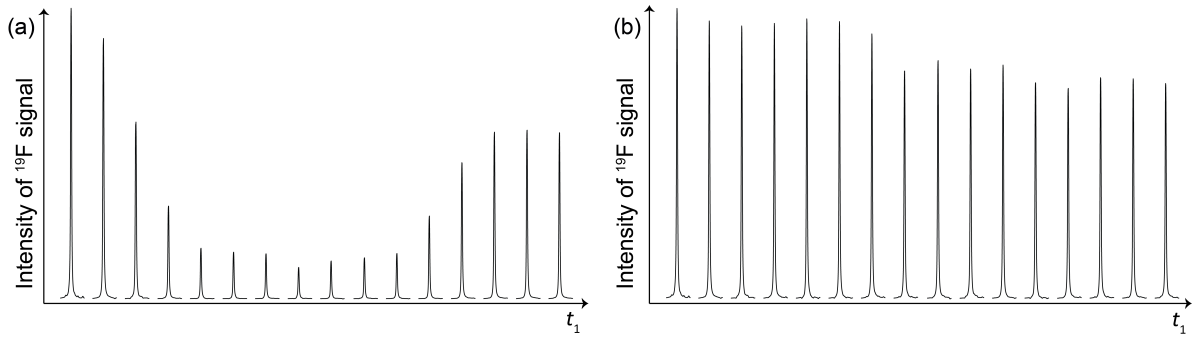

Fig. S2. Effect of ^{1}H decoupling during evolution on the AA'A''XX'X'' spin system of a solution of 10 % (v/v) $C_6F_3H_3$ in DMSO-d_6. (a) Without ^{1}H decoupling, and (b) with ^{1}H decoupling during the evolution period.

The data were acquired under similar conditions to Fig. 1 of the main text, in 30 min, using 16 transients, 4 dummy scans, 16 increments and a relaxation delay (d1) of 5 s. A Lorentzian line broadening of 0.8 Hz was used in the processing.

5. **Suppressing decoupling sidebands**

Bilevel adiabatic decoupling was used during acquisition in order to reduce decoupling sidebands to well below 0.01 %. A comparison of sideband suppression performance between the bi-level adiabatic and WALTZ-16 decoupling is shown in the following figure.

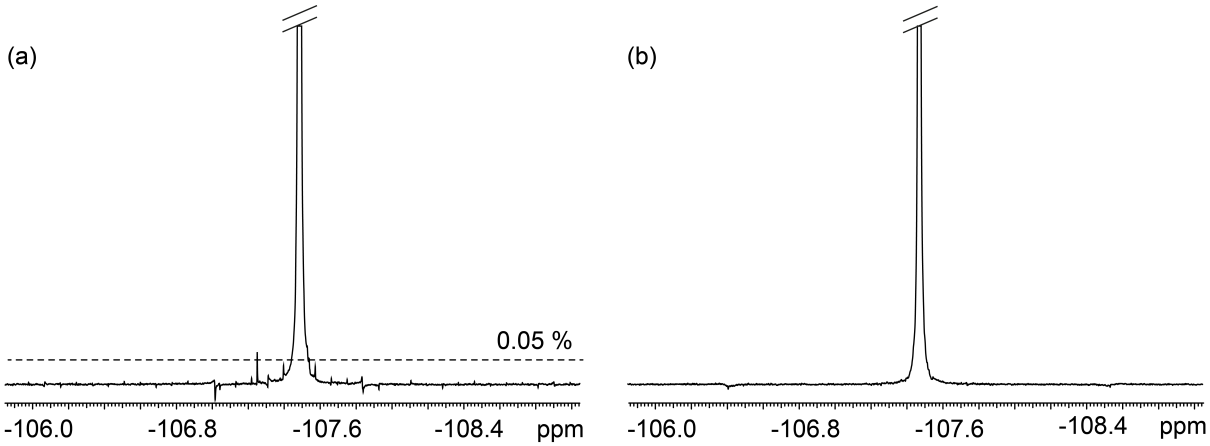

Fig. S3. Proton decoupled, ^{13}C isotopomer-suppressed ^{19}F spectra acquired with (a) WALTZ-16 and (b) bi-level ^{1}H decoupling during acquisition, for $C_6F_3H_3$ in DMSO-d_6.

The spectra were acquired with 16 transients and 32 increments using a 5 kHz spectral window in the direct dimension and a 45 Hz spectral window in the indirect dimension. 16384 complex data points were used, with zero-filling to 65536 complex points in the direct dimension and to 64 complex points in the indirect dimension.

Spectrum (a) was acquired with WALTZ-16 decoupling during acquisition and evolution, using a 50 kHz timesharing rate. Spectrum (b) used adiabatic bilevel decoupling during an explicit acquisition with a 5 kHz spectral width and WALTZ-16 decoupling during the evolution period.

6. **Conventional ^{19}F spectrum of the mixture**

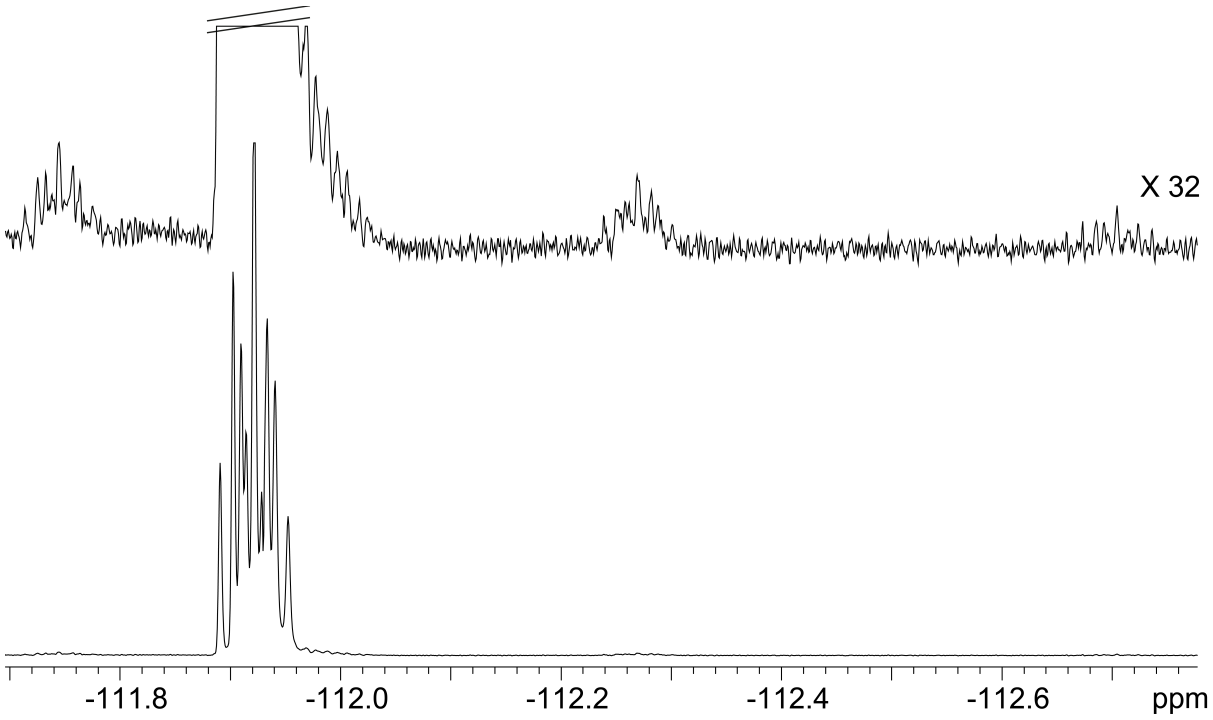

Fig. S4. 470 MHz ^{19}F spectra of the solution of Table S2 without ^{1}H decoupling or ^{13}C isotopomer suppression.

7. **Filtered heteronuclear 2D *J* spectrum**

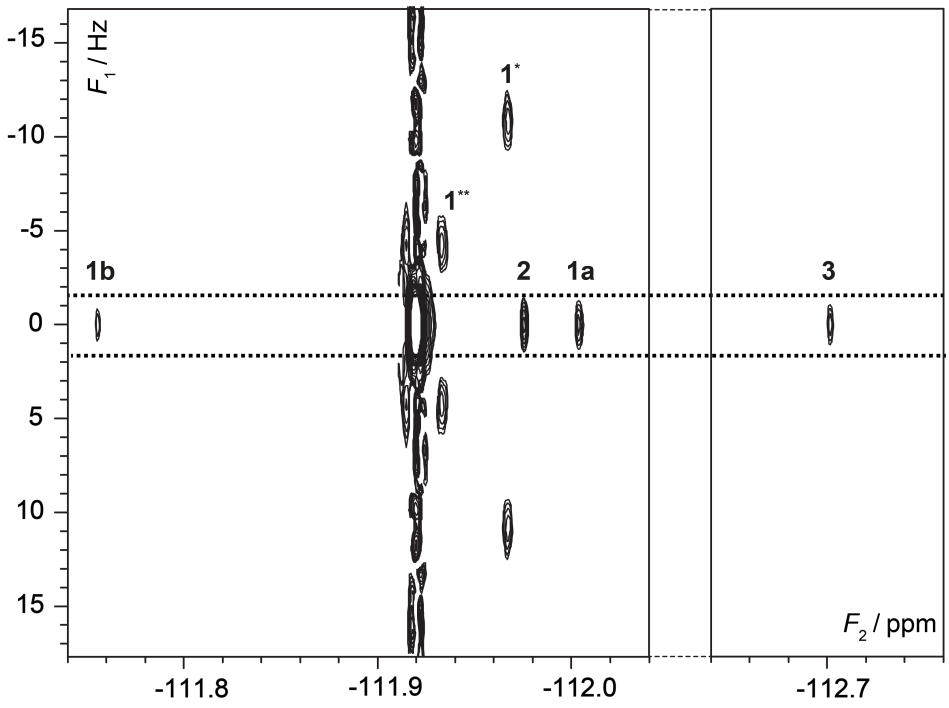

Fig S5. $^{1}J_{CF}$-suppressed, ^{19}F 2D heteronuclear *J* spectrum of a solution containing rosuvastatin (**1**), BEM (**2**), DPPO (**3**), a diastereomeric impurity in the bulk drug substance (**1a**), and a degradation impurity (**1b**), acquired with the pulse sequence of Fig. 2. Long-range ^{13}C satellite signals are denoted **1*** and **1****.

8. **Sum projection of the fresh sample**

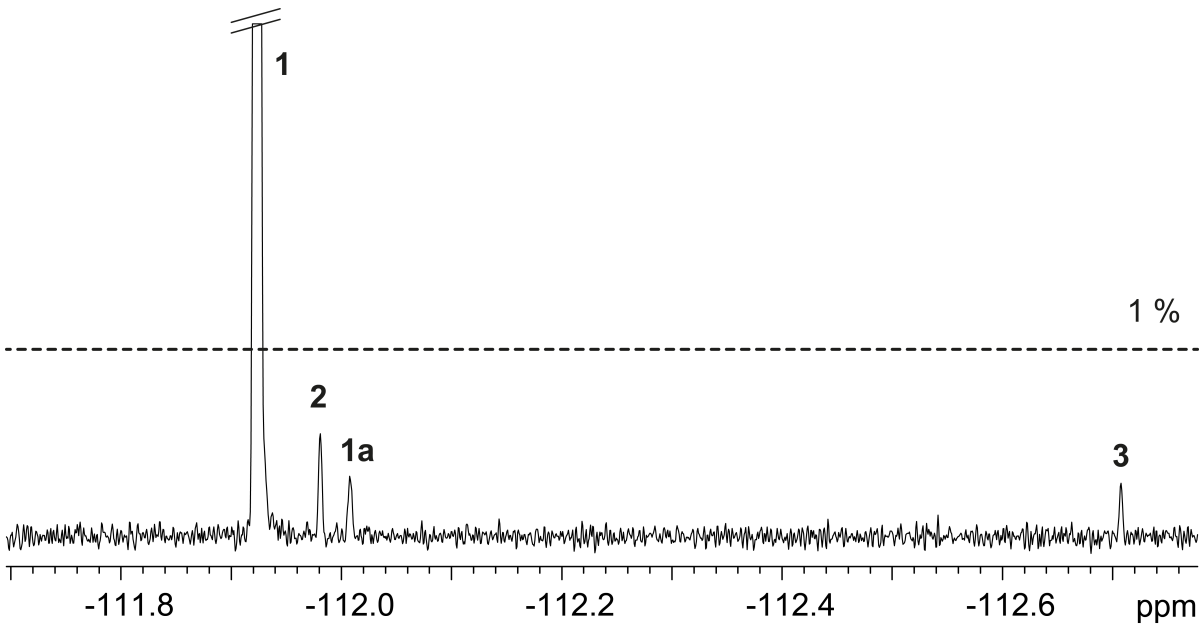

Fig. S6. Proton decoupled, ^{13}C isotopomer-suppressed ^{19}F spectrum extracted as a sum projection from F_1 = -1.7 to F_1 = +1.7 Hz of a 2D J spectrum acquired with the pulse sequence of Fig. 2 applied to a fresh sample containing the pure drug substance rosuvastatin (**1**) as a bulk component, the two precursors BEM (**2**) and DPPO (**3**) and a distereomeric impurity (**1a**).

(Here WALTZ-16 decoupling, rather than WURST-2, was used during the evolution period.)

9. Pulse sequence code

/*Pulse sequence code for the acquisition of:
1) 1H decoupled 19F spectra
2) 1H decoupled, 13C filtered and decoupled, 19F spectra
3) 1H decoupled, 13C filtered, 19F spectra */

/*----------
Developed by the NMR Methodology Group
School of Chemistry
University of Manchester
United Kingdom
July 2016
----------*/

/*User's Guide for experimental setup

1. bilevel = 'y' selects the application of the 1H adiabatic bi-level decoupling
   Note that setting dm=’nnn’ is required to avoid conflicts with default decoupling control. The decoupling pulse shapes are automatically created through the pulse sequence. Execute go(‘check’) before running the experiment, when running the experiment for the first time.
   The parameters that may need optimisation are bandwidth, pp, pplvl, and npoints. The parameter npoints together with the spectral width (sw) determines the duration of the WURST pulses. The length of the high level decoupling pulse is npoints/sw, and the low level pulse will be 2*npoints/sw. The reference 90 pulse length and power levels used by Pbox are set by the parameters pp and pplvl.
   The macro kp_readbilevelshapes is essential to run the pulse sequence.
2. satsupp = 'n' selects the acquisition of a 1D 19F
3. satsupp = 'n' and bilevel='y' selects the acquisition of a 1D 1H decoupled 19F spectrum
4. satsupp = 'y' selects the filtration of 13C satellites
   Two options within satsupp = 'y' are:
   jfilter = 'n' the short version of the pulse sequence is applied. In this case a 13C decoupling, aiming to suppress only the long range 13C satellites, should be applied using dm2=’nny’ and setting the relevant parameters: dres2, dpwr2, dmf2, dseq2, dmmp=’ccp’. The j1xh parameter should be set to a value close to $1/2(^1J_{FC})$.
   jfilter = 'y' the extended version of the pulse sequence is applied. In this case a 2D experiment is acquired. No 13C decoupling is applied. The j1xh parameter should be set to a value close to $1/2(^1J_{FC})$

The parameters preAmpConfig and probeConnect must be set properly to share a single high-band RF amplifier for proton and fluorine pulses. In our spectrometer the standard values are probeConnect=’H1 C13 N15’ preAmpConfig=’hln’, which needed to be changed to probeConnect=’F19 H1 C13’ and preAmpConfig=’hhn’.
Other spectrometer configurations may need different settings. Only channels 1 and 2 can be shared for the highband operation or alternatively channel 3 with 4 (but this would be rather unusual).
*/
/*

General Parameters

- pw : 19F 90 degree pulse width
- tpwr : 19F pulse power
- pwxlvl : 13C pulse power
- pwx : 13C 90 degree pulse width
- dpwr : decoupler power on the first decoupling channel when bi-level is not used
- dpwr2 : decoupler power on the second decoupling channel
- j1xh : One-bond fluorine-carbon coupling constant
- d1 : relaxation delay
- d2 : t_1 evolution delay (normally zero)
- gt1 : duration of the pulsed field gradient

- gzlvl1 : amplitude of the pulsed field gradient
- gstab : gradient recovery delay
- dutyc : sets the time used for decoupling in the time sharing (usually between 0.1 and 0.6)
- dbcomp : it is only needed when a conventional decoupling is used, dm='nyy', and dutyc is applied during acquisition. Then compensation should be done for the power of the decoupling during evolution where the dutyc is not applied.

Bi-level parameters

- homorof1, homorof2, homorof3: delays for the receiver's switch between on and off (typically 1-2 μs)
- npoints: npoints together with the spectral width determines the duration of the WURST pulses (see above).
- bandwidth: sets the bandwidth of the 1H decoupling
- pp: sets the 1H reference 90 degree pulse width for the creation of the WURST pulses
- pplvl: sets the reference power of the 90 degree 1H pulse for the creation of the WURST pulses
- hidseq, hidpwr, hidmf: are automatically set and read from the parameters by the pulse sequence using the macro kp_readbilevelshapes. They refer to the shape, the power and the modulation frequency of the high power WURST-2 period.
- lodseq,hidpwr,lodmf: are automatically set and read from the parameters by the pulse sequence using the macro kp_readbilevelshapes. They refer to the shape, the power and the modulation frequency of the low power WURST-2 period.
- t1dseq: are automatically set and read from the parameters by the pulse sequence using the macro kp_readbilevelshapes. They refer to the shape, the power and the modulation frequency of the decoupling applied during the evolution period.

*/

/*

Real time variables

v1-v6 : reserved for the the bi-level
v21,v22 : reserved for the bilevel loop counters
v31-v37 : reserved for phase cycling

Note: using il='y' is not compatible with this pulse sequence code, but interleaved 2D acquisition can be done by acquiring a set of 2D experiments and adding them together after acquisition, or alternatively all initval functions must be replaced by the function F_initval.

*/

```
#include <standard.h>

static int
        ph1[16] = {0,0,0,0, 1,1,1,1, 2,2,2,2, 3,3,3,3},      //v31; 1st 90 obs
        ph2[16] = {0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0},      //v32; zero; mqfilter 180 obs
        ph3[16] = {0,0,0,0, 2,2,2,2, 0,0,0,0, 2,2,2,2},      //v33; jfilter 180 obs
        ph4[16] = {0,2,0,2, 0,2,0,2, 0,2,0,2, 0,2,0,2},      //v34; 1st 90 on carbon
        ph5[16] = {0,0,2,2, 0,0,2,2, 0,0,2,2, 0,0,2,2},      //v35; 2nd 90 on carbon
        ph6[16] = {0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0},      //v36; zero 180 on carbon; in jfilter
        ph7[16] = {0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0},      //v37; zero last 90 on carbon; in jfilter

      ph8mq[16] = {0,0,0,0, 3,3,3,3, 2,2,2,2, 1,1,1,1},      //oph if jfilter='n';
        ph8[16] = {0,0,0,0, 1,1,1,1, 2,2,2,2, 3,3,3,3},      //oph if jfilter='y';
      ph8sp[16] = {0,1,2,3, 0,1,2,3, 0,1,2,3, 0,1,2,3};      //oph if satsupp='n';
```

```
pulsesequence()
{
double    j1xh   = getval("j1xh"),
          pwxlvl = getval("pwxlvl"),
          pw     = getval("pw"),
          pwx    = getval("pwx"),
          gt1    = getval("gt1"),
          gzlvl1 = getval("gzlvl1"),
          gstab  = getval("gstab"),
          tau    = 1.0/(2.0*(getval("j1xh")));

int       kpph   = getval("kpph");

char      lkgate_flg[MAXSTR],
          jfilter[MAXSTR];

          getstr("jfilter",jfilter);
          getstr("lkgate_flg",lkgate_flg);

/* variables for bilevel */

double
          dutyc       = getval("dutyc"),
          homorof1 = getval("homorof1"),
          homorof2 = getval("homorof2"),
          homorof3 = getval("homorof3"),
          dectime    = dutyc/sw,
          acqtime    = (1.0-dutyc)/sw -homorof1 - homorof2 - homorof3 - POWER_DELAY,

          npoints     = getval("npoints"),

          pw_hidec = 0.01,    //pulse length in the high level part

          pw_lodec = 0.002,  //pulse length in the low level part

          bandwidth = getval("bandwidth"),
          pwr_dec    = getval("pplvl"),
          pw90_dec = getval("pp")*1000000,


          histepsize = 5.0, lostepsize = 10.0;

      npoints = (double)( (int)(npoints));

char      bilevel[MAXSTR],
          satsupp[MAXSTR],
          cmd[MAXSTR],
          hidseq[MAXSTR],
          lodseq[MAXSTR],
          t1dseq[MAXSTR];

          getstr("bilevel",bilevel);
          getstr("satsupp",satsupp);
          getstr("hidseq",hidseq);
          getstr("lodseq",lodseq);
          getstr("t1dseq",t1dseq);



if ((npoints/sw*1000000 / histepsize) > 1000)
{
 histepsize = histepsize*(npoints/sw*1000000 / histepsize / 1000);
 printf("Default stepsize (5us) was increased to %.2f us\n",histepsize);
```

```
}
if ((npoints/sw*1000000 / lostepsize) > 500)
{
 lostepsize = lostepsize*(npoints/sw*1000000 / lostepsize / 500);
 printf("Default stepsize (10us) was increased to %.2f us\n",lostepsize);
}

if (ix==1)
{
printf("Pulse length for bilevel high decoupling was set to %.3f ms\n",npoints/sw*1000.0);
printf("Pulse length for bilevel low decoupling was set to %.3f ms\n",npoints/sw*2.0*1000.0);
}

if (dectime<0.00001)
{
abort_message("dectime is too short..");
psg_abort(1);
}
if (acqtime<0.000001)
{
abort_message("dectime is too short..");
psg_abort(1);
}
if ( npoints/sw < 0.001)
{
printf("Set npoints more than %0.f or change sw..\n",sw*0.001);
abort_message("npoints is too small..");
psg_abort(1);
}
if ( npoints/sw*20 > at)
{
abort_message("Reduce npoints or increase at..");
psg_abort(1);
}


if (ix==1)
{
//make decoupler shapes
        sprintf(cmd,"Pbox kp_highdec.DEC -u %s -w \"wurst2i %.1f/%.7f\" -sucyc t5 -s %.1f -dcyc %.2f -p %.0f -l %.1f",
userdir, bandwidth, npoints/sw , histepsize, dutyc, pwr_dec, pw90_dec);
        system(cmd);
        sprintf(cmd,"Pbox kp_lowdec.DEC -u %s -w \"WURST2 %.1f/%.7f\" -sucyc m16 -s %.1f -dcyc %.2f -p %.0f -l
%.1f", userdir, bandwidth, npoints/sw*2.0, lostepsize, dutyc, pwr_dec, pw90_dec);
        system(cmd);
        sprintf(cmd,"Pbox kp_t1dec.DEC -u %s -w \"WURST2 %.1f/%.7f\" -sucyc m16 -s %.1f -p %.0f -l %.1f", userdir,
bandwidth, npoints/sw*2.0, lostepsize, pwr_dec, pw90_dec);
        system(cmd);
  putCmd("kp_readbilevelshapes\n");
}


double  hidpwr = getval("hidpwr"),
        hidmf  = getval("hidmf"),
        hidres = getval("hidres"),
        lodpwr = getval("lodpwr"),
        lodmf  = getval("lodmf"),
        lodres = getval("lodres"),
        t1dpwr = getval("t1dpwr"),
        t1dmf  = getval("t1dmf"),
        t1dres = getval("t1dres");


if ((hidpwr>56) || (lodpwr>50))
{
abort_message("Set npoints is too small; too much power on decoupler. Increase npoints or dutyc..");
psg_abort(1);
```

```
}


//phase cycle
//v5: low dec cycle
//v1: high dec cycle
        assign(two,v3);
        mult(v3,v3,v3); mult(v3,v3,v3);
        assign(ct,v1);
        modn(v1,v3,v1);
        assign(one,v2);     add(v1,v2,v1);
        initval(npoints,v6); mult(v6,v1,v1);
        sub(v3,v1,v5);
        initval(np/2-16,v4);
        add(v4,v5,v5);


  assign(ct,v17);

if (kpph==0)
{
   settable(t1,16,ph1);
   settable(t2,16,ph2);
   settable(t3,16,ph3);
   settable(t4,16,ph4);
   settable(t5,16,ph5);
   settable(t6,16,ph6);
   settable(t7,16,ph7);

        if (satsupp[0]=='n')
        {
        settable(t8,16,ph8sp);
        }
        if (jfilter[0]=='n')
        {
        settable(t8,16,ph8mq);
        }
        if ((jfilter[0]=='y') && (satsupp[0]=='y'))
        {
        settable(t8,16,ph8);
        }

}
else
{
   settable(t1,kpph,ph1);
   settable(t2,kpph,ph2);
   settable(t3,kpph,ph3);
   settable(t4,kpph,ph4);
   settable(t5,kpph,ph5);
   settable(t6,kpph,ph6);
   settable(t7,kpph,ph7);
        if (satsupp[0]=='n')
        {
        settable(t8,kpph,ph8sp);
        }
        if (jfilter[0]=='n')
        {
        settable(t8,kpph,ph8mq);
        }
        if ((jfilter[0]=='y') && (satsupp[0]=='y'))
        {
        settable(t8,kpph,ph8);
        }

}
```

```
  getelem(t1, v17, v31);
  getelem(t2, v17, v32);
  getelem(t3, v17, v33);
  getelem(t4, v17, v34);
  getelem(t5, v17, v35);
  getelem(t6, v17, v36);
  getelem(t7, v17, v37);
  getelem(t8, v17, oph);



/* equilibrium period */

   status(A);

   obspower(tpwr);
   decpower(dpwr - getval("dbcomp") );
   dec2power(pwxlvl);

   delay(0.02);
   if ( (lkgate_flg[0] == 'y') || (lkgate_flg[0] == 'k') )  lk_sample(); // turn on lock sampling
   delay(d1);
   if ( (lkgate_flg[0] == 'y') || (lkgate_flg[0] == 'k') )  lk_hold(); // turn off lock sampling
   delay(0.02);


if (satsupp[0]=='y')
{

  if  (jfilter[0]=='n')
  {
status(B);
  delay(0.05);
  rgpulse(pw,v31,rof1,rof1);

   delay(tau-gt1-gstab);
   zgradpulse(gzlvl1,gt1);
   delay(gstab);
   rgpulse(2.0*pw,v32,rof1,rof1);
   dec2rgpulse(pwx,v34,rof1,rof1);
   zgradpulse(gzlvl1,gt1);
   delay(gstab);
   delay(tau-pwx-gt1-gstab);
   dec2rgpulse(pwx,v35,rof1,rof2);
  }
  else
  {
              if (bilevel[0]=='y')
              {
                decpower(t1dpwr);
              }
status(B);
              if (bilevel[0]=='y')
              {
                decprgon(t1dseq, 1.0/t1dmf, t1dres);
              }
   delay(0.05);
   rgpulse(pw,v31,rof1,rof1);

   delay(tau-gt1-gstab);
   zgradpulse(gzlvl1,gt1);
   delay(gstab);
   rgpulse(2.0*pw,v32,rof1,rof1);
   dec2rgpulse(pwx,v34,rof1,rof1);
   zgradpulse(gzlvl1,gt1);
   delay(gstab);
```

```
   delay(tau-pwx-gt1-gstab);
   dec2rgpulse(pwx,v35,rof1,rof1);
   delay(d2/2.0);
   rgpulse(2.0*pw,v33,rof1,rof1);
   dec2rgpulse(2.0*pwx,v36,rof1,rof1);
   delay(d2/2.0);
   dec2rgpulse(pwx,v37,rof1,rof2);

              if (bilevel[0]=='y')
              {
                decoff(); decpower(-16.0); decblank();
                decprgoff();
              }
  }
//end of satsupp flag
}
else
{

status(B);
delay(d2);
   rgpulse(pw,oph,rof1,rof2);
}




if (bilevel[0]=='y')
{


   /* --- observe period --- */

dec2power(dpwr2);
status(C);


//high level part
              decpower(hidpwr);
decprgon(hidseq, 1.0/hidmf, hidres);

              startacq(alfa);
  loop(v1,v21);
              decoff(); decpower(-16.0); decblank();
              delay(homorof3);
              rcvron();

              delay(homorof1);
              acquire(2.0, acqtime);
              rcvroff();
              delay(homorof2);
              decpower(hidpwr); decon(); decunblank();
              delay(dectime);
  endloop(v21);
              decoff(); decpower(-16.0); decblank();
decprgoff();

//low level part
              decpower(lodpwr);
decprgon(lodseq, 1.0/lodmf, lodres);


  loop(v5,v22);
              decoff(); decpower(-16.0); decblank();
              delay(homorof3);
              rcvron();
```

```
            delay(homorof1);
            acquire(2.0, acqtime);
            rcvroff();
            delay(homorof2);
            decpower(lodpwr); decon(); decunblank();
            delay(dectime);
  endloop(v22);


            decoff(); decpower(-16.0); decblank();
            delay(homorof3);
decprgoff();

 if (lkgate_flg[0] == 'y') lk_sample();

}
else
{
  /* --- observe period --- */

decpower(dpwr);
dec2power(dpwr2);
status(C);

 if (lkgate_flg[0] == 'y') lk_sample();

}




}
```