Supporting information:

An enhanced sensitivity methyl ^{1}H triple-quantum pulse scheme for measuring diffusion constants of macromolecules

Rui Huang[1], Jacob P. Brady[1], Ashok Sekhar[1], Tairan Yuwen[1] and Lewis E. Kay[1,2]

[1]Departments of Molecular Genetics, Biochemistry and Chemistry, University of Toronto, Toronto, Ontario, M5S 1A8, Canada.

[2]The Hospital for Sick Children Research Institute, Toronto, Ontario, M5G 0A4, Canada.

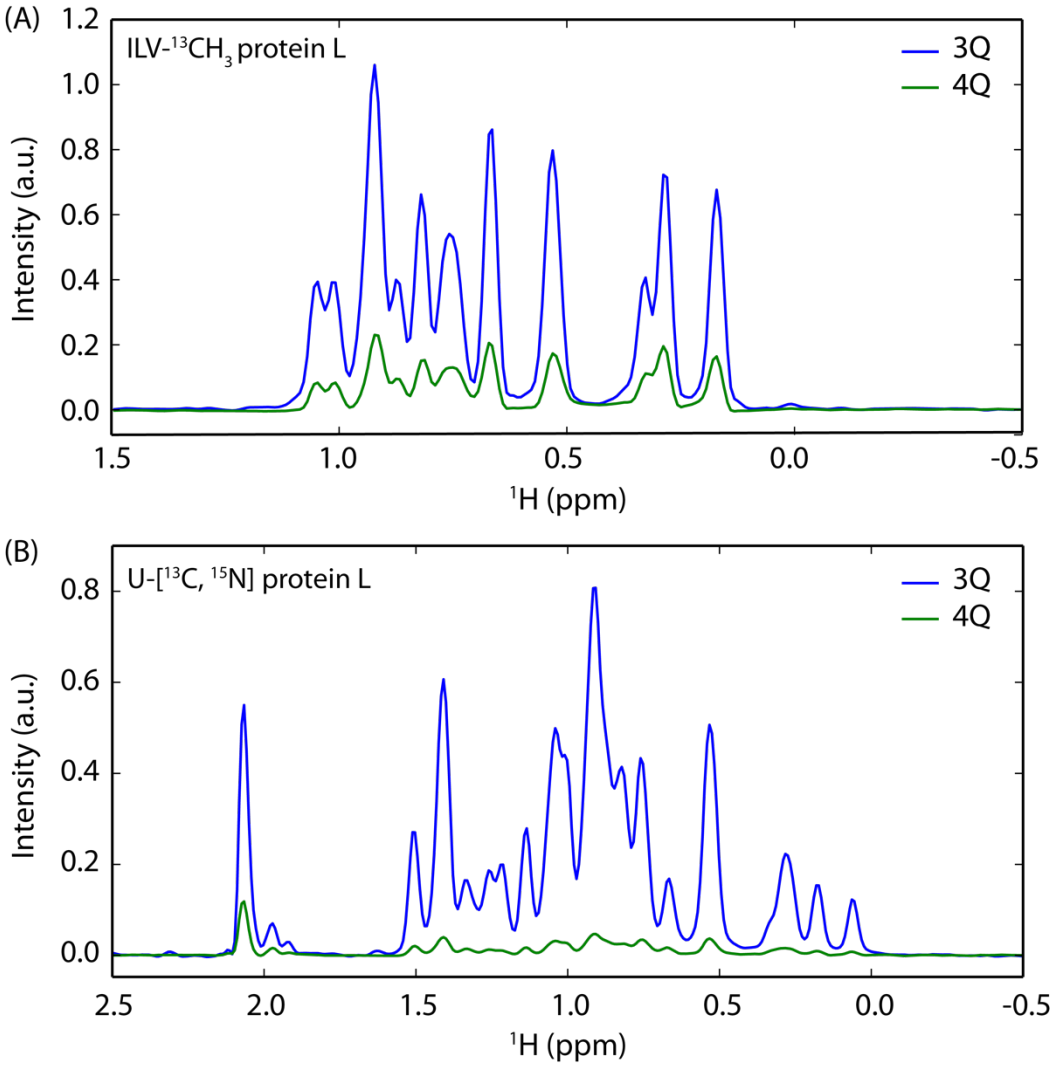

Figure S1. Comparison of 1D spectra obtained with triple-quantum (3Q, blue)- and 4-quantum (4Q, green)-based (Zheng et al. 2009) diffusion experiments recorded on samples of ^{2}H, Ileδ1-[^{13}CH$_3$],Leu,Val-[^{13}CH$_3$,^{12}CD$_3$]-labeled protein L (A) and a U-[^{13}C,^{15}N]-labeled protein L (B), 25 ^{o}C. Gradient strengths of the encoding/decoding gradients were set to zero, and Δ' and $\delta/2$ were set to 0.02 s and 0.3 ms, respectively. Identical numbers of scans were recorded for each of the

compared datasets. Note that in (B) the losses are smaller for the Met peaks because these derive from methyl groups whose carbons are not one-bond coupled to ^{13}C spins.

References

Zheng G, Torres AM, Price WS (2009) MQ-PGSTE: A new multi-quantum STE-based PGSE NMR sequence. J Magn Reson 198:271–274. doi: 10.1016/j.jmr.2009.03.004

```
/* hsqc_1D_3Q_diff_600_lek_cp

  Used to record 1D diffusion using 13CH3 methyl by creating 1H 3Q
    Magnetization is of the form and IZCZ during diff time

 Set list of gradient multipliers in Diff_3Q file in gp directory under list

  Written by LEK on Jan 29, 2016

  72 step phase cycle

  Modified by LEK on Feb 24, 2016 to select for specific methyls in cases where there
are
    peaks from CH2 that breakthrough

  Modified by LEK on April 9, 2017 to include the option of having magnetization   of
the form Iz and IzIzIz during diffusion delay

  Modified by LEK on April 20, 2017 to include water gate (-Dw_gate)
*/

#include <Avance.incl>
#include <Grad.incl>
#include <Delay.incl>

;Define phases
#define zero ph=0.0
#define one ph=90.0
#define two ph=180.0
#define three ph=270.0

;Define Pulses
define pulse pwh
       "pwh=p1"              ; 1H hard pulse at power pl1
define pulse pwc
       "pwc=p2"              ; 13C hard pulse at power pl2

#ifndef C_sel
define pulse pwc_ad          ; adiabatic pulse for inept transfers
       "pwc_ad = p23"
#endif

#ifdef C_sel
define pulse pwc_reb          ; selective reburp pulse for inept transfers
       "pwc_reb = p24"
#endif

;Define delays

"in0=inf1/2"
"d11=30m"

define list<gradient> diff=<Diff_3Q>

define delay taua
       "taua=d5"             ; < 1/(4JCH) 1.8 ms

define delay taub
       "taub=d6"             ; = 1/(4JCH) 2.0 ms

define delay T_diff
       "T_diff = d9"

define delay hscuba
  "hscuba=30m"

/* Assign cnsts to check validity of parameter changes */
#ifdef fsat
   "cnst10 = plw10" ; tsatpwr - set to max of 0.00005W
```

```
#endif

   "cnst21=plw21" ; dpwr pl21 - set max at 2.5W

#ifdef C_sel
    "cnst24 = spw24"  ; power level for selective 13C reburp
#endif

#ifndef w_gate
"acqt0=-pwh*2.0/PI - 2u"  ; select 'DIGIMOD = baseopt' to execute
#else
"acqt0=-2u"  ; select 'DIGIMOD = baseopt' to execute
#endif

/*************************************/
/* Define macro for 3-9-19 WATERGATE */
/*************************************/
#ifdef w_gate
#define Watergate (center (pwh*0.231 ph26 d19*2 pwh*0.692 ph28 d19*2 pwh*1.462 ph26 d1
9*2 pwh*1.462 ph26 d19*2 pwh*0.692 ph28 d19*2 pwh*0.231 ph26):f1 (pwc*2 ph26):f2)
#endif

1 ze

; check validity of parameters

#ifdef fsat
  if "cnst10 > 0.001"
  {
   2u
   print "error: presat power pl10 is too large"
   goto HaltAcqu
  }
#endif

  if "ns % 72 !=0"
 {
   2u
   print "error: ns must be a multiple of 72"
   goto HaltAcqu
  }

  if " cnst21 > 2.5"
  {
   2u
   print "error: dpwr pl21 too large"
   goto HaltAcqu
  }

#ifdef C_sel
  if " cnst24 > 2.0"
  {
   2u
   print "error: selective reburp power too large"
   goto HaltAcqu
  }
#endif

  if "pwc > 20u"
  {
   2u
   print "error: pwc too large < 20 us"
   goto HaltAcqu
  }

#ifndef C_sel
  if "pwc_ad > 500u"
  {
   2u
```

```
   print "error: pwc_ad too large < 500 us"
   goto HaltAcqu
  }
#endif

#ifdef C_sel
  if "pwc_reb < 2000u"
  {
   2u
   print "error: pwc_reb too small < 2000 us"
   goto HaltAcqu
  }
#endif

2 d11 do:f2

#ifdef scramble
; scramble any residual 1H magnetization

 20u UNBLKGRAD          ; dly 20u, unblank gradients and lock hold

 2u pl1:f1
 (pwh one):f1

  2u
  p56:gp6*0.25       ; gradient 6 * 0.25
  d16

 2u pl1:f1
 (pwh zero):f1

  2u
  p56:gp6*0.5       ; gradient 6*0.5
  d16

 20u BLKGRAD          ; dly 20u, blank gradients

#endif

#ifdef fsat                          ; zgoptn -Dfsat
20u fq=cnst1:f1                      ; jump from o1 to water
  4u pl10:f1                         ; power(tsatpwr) for presaturation
  d1 cw:f1 zero                      ; Hcw(d1)x
  4u do:f1                           ; cw off
  2u pl1:f1                          ; power(tpwr)

#ifdef fscuba                    /* Scuba pulse sequence */
   hscuba                           ; delay(hscuba)
    (pwh zero):f1                   ; H 90x180y90x
    (pwh*2 one):f1
    (pwh zero):f1
   hscuba                           ; delay(hscuba)
#endif                            /* end fscuba */

#else                              /* if fsat is no */
 2u pl1:f1                         ; power(tpwr)
 d1                                ; delay(d1)
#endif                             /* end if fsat */

  2u pl1:f1
  2u pl2:f2
  2u pl31:f3

 20u UNBLKGRAD           ; dly 20u, unblank gradients and lock on hold

 10u fq=0:f1         ; jump back to o1

  (pwc zero):f2   ; C90x - To destroy 13C Boltzman magnetization
```

```
  2u
  (p50:gp0)       ; gradient 0
  d16

; This is the real start

(pwh ph1):f1                              ; H90x

2u
p51:gp1                                   ; gradient 1
d16

#ifndef C_sel
"DELTA = taua - 2.0u - p51 - d16 - pwc_ad*0.5 - pwh*4/PI - 0.2u" ; subtract effects of
 both 1H 90
DELTA                                     ; delay 1/4JCH

(center(pwh*2 ph1):f1 (pwc_ad:sp23 ph26):f2)  ; H180x,C180x

"DELTA = taua - 2.0u - p51 - d16 - pwc_ad*0.5 - 2u"
DELTA                                     ; delay 1/4JCH
#endif

#ifdef C_sel
"DELTA = taua - 2.0u - p51 - d16 - pwc_reb*0.5 - pwh*4/PI - 0.2u" ; subtract effects o
f both 1H 90
DELTA                                     ; delay 1/4JCH

(center(pwh*2 ph1):f1 (pwc_reb:sp24 ph26):f2)  ; H180x,C180x

"DELTA = taua - 2.0u - p51 - d16 - pwc_reb*0.5 - 2u"
DELTA                                     ; delay 1/4JCH
#endif

2u
p51:gp1                                   ; gradient 1
d16

2u pl2:f2
(pwc ph2):f2                               ; C90ph1

2u
p61:gp11                                   ; gradient 11
d16

"DELTA = taub - 2.0u - p61 - d16"
DELTA                                     ; delay 1/4JCH

(center(pwh*2 ph1):f1 (pwc ph27 pwc*2 ph26 pwc ph27):f2)  ; H180x,C180x

"DELTA = taub - 2.0u - p61 - d16"
DELTA                                     ; delay 1/4JCH

2u
p61:gp11                                  ; gradient 11
d16

(pwc zero):f2                               ; C90x
0.2u
(pwh ph1):f1                              ; H90t2

2u
p53:gp3*diff                              ; gradient 3 diffusion encoding gradient
500u

(pwh ph27 pwh*2 ph26 pwh ph27):f1

2u
p53:gp3*-1.0*diff                          ;  - gradient 3 diffusion encoding gradient
```

```
500u

#ifndef Iz     ; magnetization diffuses as 2IzCz

(pwh ph3):f1                              ; H90ph3
1.0u
(pwc zero):f2                             ; C90x

2u
p54:gp4                                   ; gradient 4
d16

"DELTA = taub - 2.0u - p54 - d16"
DELTA                                     ; delay 1/4JCH

(center(pwh ph27 pwh*2 ph26 pwh ph27):f1 (pwc ph27 pwc*2 ph26 pwc ph27):f2)  ; H180x,C
180x

"DELTA = taub - 2.0u - p54 - d16"
DELTA                                     ; delay 1/4JCH

2u
p54:gp4                                  ; gradient 4
d16

(pwc zero):f2                               ; C90x
1.0u
(pwh one):f1

#else

(pwh ph3):f1                              ; H90ph3
1.0u
(pwc zero):f2                             ; C90x

2u
p54:gp4                                   ; gradient 4
d16

"DELTA = taub - 2.0u - p54 - d16"
DELTA                                     ; delay 1/4JCH

(center(pwh ph27 pwh*2 ph26 pwh ph27):f1 (pwc ph27 pwc*2 ph26 pwc ph27):f2)  ; H180x,C
180x

"DELTA = taub - 2.0u - p54 - d16"
DELTA                                     ; delay 1/4JCH

2u
p54:gp4                                  ; gradient 4
d16

(pwc zero):f2
1.0u                                     ; compensates for the 1 us above

2u
p54:gp4*0.5                                 ; gradient 4
d16

"DELTA = taub - 2.0u - p54 - d16"
DELTA                                     ; delay 1/4JCH

(center(pwh ph27 pwh*2 ph26 pwh ph27):f1 (pwc ph27 pwc*2 ph26 pwc ph27):f2)  ; H180x,C
180x

"DELTA = taub - 2.0u - p54 - d16 - pwh*2.0/PI"
DELTA                                     ; delay 1/4JCH

2u
```

```
p54:gp4*0.5                                ; gradient 4
d16

(pwh zero):f1
#endif

#ifndef Iz     ; magnetization diffuses as 2IzCz
"DELTA = T_diff - taub*4 - 2u - p56 - d16"
#else
"DELTA = T_diff - taub*8 - 2u - p56 - d16"
#endif

#ifndef fsat_diff
DELTA
#endif

#ifdef fsat_diff                      ; zgoptn -Dfsat
  10u fq=cnst1:f1                     ; jump from o1 to water
  2u pl10:f1                          ; power(tsatpwr) for presaturation
  DELTA cw:f1 zero                    ; Hcw(d1)x
  2u do:f1                            ; cw off
  2u pl1:f1                           ; power(tpwr)
  10u fq=0:f1                         ; jump back to o1
#endif

2u
p56:gp6                                ; gradient 6
d16

#ifndef Iz

(pwh ph4):f1
1.0u
(pwc ph5):f2                             ; C90x

  2u
  p55:gp5                                ; gradient 5
  d16

 "DELTA = taub - 2.0u - p55 - d16"
  DELTA                                  ; delay 1/4JCH

  (center(pwh*2 ph4):f1 (pwc ph27 pwc*2 ph26 pwc ph27):f2)  ; H180x,C180x

 "DELTA = taub - 2.0u - p55 - d16"
  DELTA                                  ; delay 1/4JCH

  2u
  p55:gp5                                ; gradient 5
  d16

(pwc zero):f2                            ; C90x
1.0u
(pwh ph6):f1

#else

(pwh ph14):f1

  2u
  p55:gp5*0.5                                ; gradient 5
  d16

 "DELTA = taub - 2.0u - p55 - d16 - pwh*2.0/PI - 1u"
  DELTA                                  ; delay 1/4JCH

  (center(pwh*2 ph14):f1 (pwc ph27 pwc*2 ph26 pwc ph27):f2)  ; H180x,C180x

 "DELTA = taub - 2.0u - p55 - d16"
```

```
  DELTA                                    ; delay 1/4JCH

  2u
  p55:gp5*0.5                                ; gradient 5
  d16

(pwc zero):f2

  2u
  p55:gp5                                  ; gradient 5
  d16

 "DELTA = taub - 2.0u - p55 - d16"
  DELTA                                    ; delay 1/4JCH

  (center(pwh*2 ph14):f1 (pwc ph27 pwc*2 ph26 pwc ph27):f2)  ; H180x,C180x

 "DELTA = taub - 2.0u - p55 - d16"
  DELTA                                    ; delay 1/4JCH

  2u
  p55:gp5                                 ; gradient 5
  d16

(pwc ph5):f2
1u
(pwh ph14):f1

#endif

  2u
  p53:gp3*diff                             ; gradient 3 diffusion encoding gradient
  500u

  (pwh ph27 pwh*2 ph26 pwh ph27):f1

  2u
  p53:gp3*-1.0*diff                         ;  - gradient 3 diffusion encoding gradien
t
  500u

(pwh ph7):f1                              ; H90ph7
1.0u
(pwc zero):f2                             ; C90x

  2u
  p58:gp8                                  ; gradient 8
  d16

 "DELTA = taub - 2.0u - p58 - d16"
  DELTA                                    ; delay 1/4JCH

  (center(pwh ph27 pwh*2 ph26 pwh ph27):f1 (pwc ph27 pwc*2 ph26 pwc ph27):f2)  ; H180x
,C180x

 "DELTA = taub - 2.0u - p58 - d16"
  DELTA                                    ; delay 1/4JCH

  2u
  p58:gp8                                 ; gradient 8
  d16

#ifndef w_gate

  (pwc zero):f2                            ; C90x

  2u
  p60:gp10                       ; gradient 10
  d16
```

```
#ifndef C_sel

 "DELTA = taua - 2.0u - p60 - d16 - pwc_ad*0.5 + pwh*2/PI"
  DELTA                                     ; delay 1/4JCH

  (center(pwh*2 ph26):f1 (pwc_ad:sp23 ph26):f2)  ; H180x,C180x

  2u pl2:f2

 "DELTA = taua - 2u - 2u - p60 - d16 - pwc_ad*0.5 - 4u - pwc - pwc - pwh*2/PI"
  DELTA                                     ; delay 1/4JCH
#endif

#ifdef C_sel
 "DELTA = taua - 2.0u - p60 - d16 - pwc_reb*0.5 + pwh*2/PI"
  DELTA                                     ; delay 1/4JCH

  (center(pwh*2 ph26):f1 (pwc_reb:sp24 ph26):f2)  ; H180x,C180x

  2u pl2:f2

 "DELTA = taua - 2u - 2u - p60 - d16 - pwc_reb*0.5 - 4u - pwc - pwc - pwh*2/PI"
  DELTA                                     ; delay 1/4JCH
#endif

  2u
  p60:gp10                                  ; gradient 10
  d16

  4u BLKGRAD

  (pwc zero):f2                            ; C90x
  (pwc ph3):f2                            ; C90x

  (pwh one):f1             ; C90(+/-x),H90y
#else  ; if w_gate is used
  (pwc zero):f2                             ; C90x

  2u
  p60:gp10                       ; gradient 10
  d16

 "DELTA = taua - 2.0u - p60 - d16 - d19*3.367 - 2u"
  DELTA
  2u
  Watergate

 "DELTA = taua - 2.0u - p60 - d16 - d19*3.367 - 4u - pwc - pwc"
  DELTA

  2u
  p60:gp10                       ; gradient 10
  d16

  4u BLKGRAD

  (pwc zero):f2                            ; C90x
  (pwc ph3):f2                            ; C90x
#endif

  2u pl21:f2               ; lower power for 13C decoupling

  go=2 ph31 cpds2:f2        ; acquire fid with syncronous 13C decoupling during aq
  d11 do:f2 mc #0 to 2 F0(zd) ; write fid to disk
  F1QF(calgrad(diff))

HaltAcqu, 1m
exit
```

```
ph1= (6) 0 1 2 3 4 5  ; 0 60 120 180 240 300
ph2= 0 0 0 0 0 0 0 0 0 0 0 0
     0 0 0 0 0 0 0 0 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 2 2 2 2 2 2 2 2
     2 2 2 2 2 2 2 2 2 2 2 2
ph3 = 0 0 0 0 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 2 2 2 2
ph4= (12) 3 3 3 3 3 3 5 5 5 5 5 5 7 7 7 7 7 7 9 9 9 9 9 9 11 11 11 11 11 11 1 1 1 1 1 1
ph5 = 0 0 0 0 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 2 2 2 2
ph6= (6) 0 0 0 0 0 0 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4 5 5 5 5 5 5
ph7=  0
ph14= (6) 0 0 0 0 0 0 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4 5 5 5 5 5 5
ph31=0 2 0 2 0 2
     2 0 2 0 2 0
     0 2 0 2 0 2
     0 2 0 2 0 2
     2 0 2 0 2 0
     0 2 0 2 0 2
     2 0 2 0 2 0
     0 2 0 2 0 2
     2 0 2 0 2 0
     2 0 2 0 2 0
     0 2 0 2 0 2
     2 0 2 0 2 0
ph26=0
ph27=1
ph28=2
ph29=3

;d1 : repetition delay
;d5 : taua ~1/4JCH
;d6 : taub = 1/4JCH 2 ms
;d9 : T_diff total diffusion time
;d11 : delay for disk i/o, 30ms
;d16 : gradient recovery delay, 200us
;d19 : delay for binomial water suppression = |1/(4*dis)| dis = Hz between o1 and wate
r
;      d19 = (1/(4*|cnst1|)), cnst1 = distance between methyl and water (in Hz)
;pl1 : tpwr - power level for pwh
;pl2 : dhpwr - power level for hard 13C pulse pwc
;pl21 : dpwr - power level for  13C decoupling cpd2
;sp23 : power level for pwc_ad
;sp24 : power level for selective reburp
;p23  : pwc_ad - adiabatic pulses for INEPT transfer
;p24 :  pwc_reb - selective reburp pulse
;pl31 : dpwr2 - power level for 15N cpd3
;pl4 : dpwr3 - power level for 2H flipback pulses
;pl41 : dpwr3D - power level for 2H cpd4
;p1 : pwh
;p2 : pwc
;p31 : pwn at dpwr2 for 15N decoupling during 2*TC(~29ms)
;p4 : pwd at dpwr3 for 2H flipback pulses
;p41 : pwddec at dpwr3D for 2H decoupling
;cpd2 : 13C decoupling according to program defined by cpdprg2
;cpd4 : 2H decoupling according to program defined by cpdprg4
;pcpd2:  13C 90 degree pulse at pl21 for cpd2
;pcpd3:  15N 90 degree pulse at pl31 for cpd3
;pcpd4:  2H 90 degree pulse at pl41 for cpd4
;spnam23 : file name for chirp pulse during inepts
;p53 : duration of diffusion encoding gradient pulse
;GPZ3 : set to 100 % for gradient encoding
;cnst1 : water - 01 (Hz)
;cnst12 : power level in W for scrambling pulse
;zgoptns : Dfsat,Dfscuba,Dfsat_diff,Dscramble,DC_sel,DIz,Dw_gate
```