Sparse Multidimensional Iterative Lineshape-Enhanced (SMILE) Reconstruction of Both Non-Uniformly Sampled and Conventional NMR Data

Jinfa Ying,[1] Frank Delaglio,[2] Dennis A. Torchia[3] and Ad Bax[1]

[1] Laboratory of Chemical Physics, National Institute of Digestive and Diabetic and Kidney Diseases, National Institutes of Health, Bethesda, MD 20892, U.S.A.

[2] Institute for Bioscience and Biotechnology Research
National Institute of Standards and Technology and the University of Maryland,
Rockville, MD 20850 U.S.A.

[3] National Institute of Dental and Craniofacial Research,
National Institutes of Health, Bethesda, MD 20892, U.S.A.

Supplementary Materials

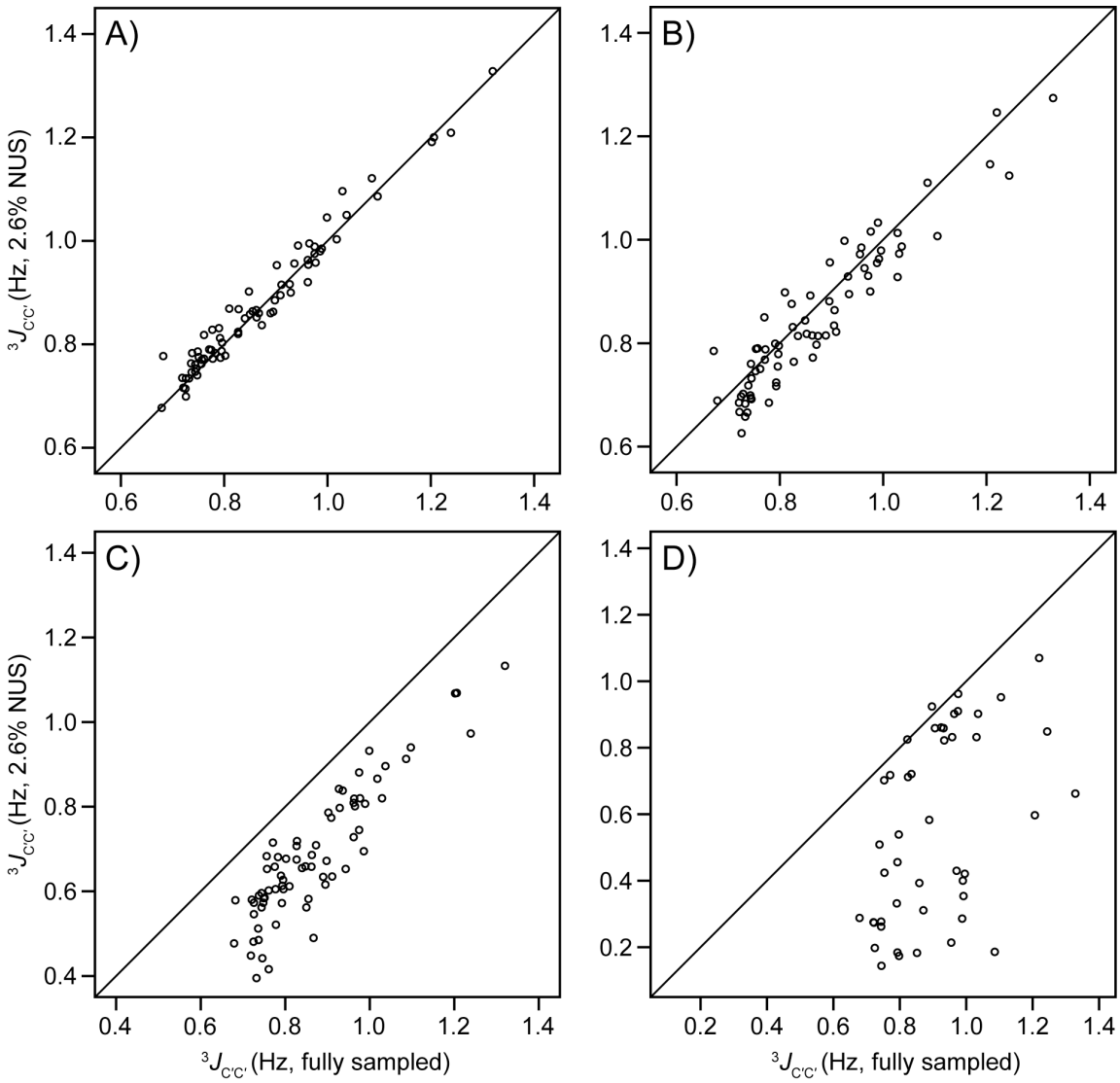

Fig. S1. Correlation plots of $^3J_{C'C'}$ couplings obtained using different NUS reconstruction programs. The same (H)N(COCO)NH NUS data set of α-synuclein with a sparsity of 2.6%, used for the data shown in Figs. 3 and 4, main text, was used as input for different reconstruction programs, and the corresponding $^3J_{C'C'}$ values are compared to these obtained from the fully sampled spectrum. (A) SCRUB (Coggins et al. 2012). Although not required for SMILE-reconstruction, the (H)N(COCO)NH was recorded with zero linear phase corrections in both ^{15}N dimensions to allow for optimal reconstruction by SCRUB; (B) SSA with the time-domain fitting option turned off (Stanek et al. 2012); (C) istHMS (Hyberts et al. 2012); and (D) IRLS in the MddNMR package (Kazimierczuk and Orekhov 2011). The intensity ratio of the intense diagonal resonances and corresponding weak cross peaks is used to calculate $^3J_{C'C'}$, and the extracted couplings are sensitive monitors for how accurately this ratio can be extracted from the NUS-reconstructed spectrum relative to the fully sampled data set. Whereas with an RMSD of 0.028 Hz, SCRUB yields essentially indistinguishable results from SMILE (main text), and slightly higher for SSA (0.044 Hz), both istHMS (RMSD 0.20 Hz) and IRLS-MddNMR (0.43 Hz) tend to systematically underestimate the intensity of the very weak resonances.

Fewer pairs of cross peaks were detectable in the SCRUB (N=73), SSA (N=73), istHMS (N=73), IRLS-MddNMR (N=46) processed spectra than in SMILE (N=74) spectra. Note that for SSA and IRLS-MddNMR generated spectra the analysis was carried out by the program Sparky (Goddard and Kneller 2008) as the SSA-processed data were kindly supplied to us in that data format, and not easily back-converted to NMRPipe format. Testing Sparky peak picking on SMILE-processed data indicates that a small fraction of the lower reproducibility of SSA and MddNMR-processed spectra likely can be attributed to the superior intensity measurement of weak peaks by NMRPipe over Sparky.

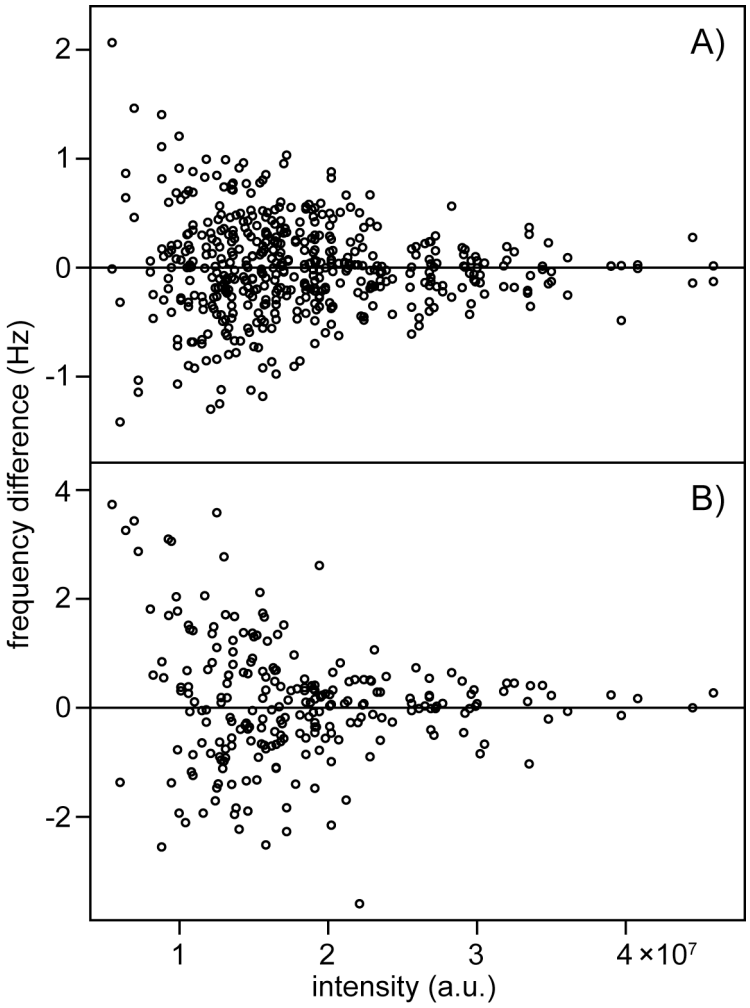

Fig. S2. Deviation of crosspeak positions in the SCRUB reconstructed spectrum from the reference fully sampled spectrum (Fig. 3A) plotted against the crosspeak intensities in the reconstructed spectrum. (A) Differences in the ^{15}N indirect dimensions and (B) in the direct ^{1}H dimension. The frequency spread (rms deviation from the fully sampled spectrum 0.438 and 1.105 Hz for ^{15}N and ^{1}H positions, respectively) are slightly larger than that obtained from SMILE and plotted in Fig. 5 of the main text. Similar results are seen in the data obtained using the other NUS reconstruction programs.

Table S1. RMS deviations in the peak position, width and height, and the average deviation in the peak height in the istHMS spectra relative to the noise-free fully sampled reference spectra.

	peak	relative height	fully sampled[a]	random[b]	T_2 weighted[b]	sine-weigthed Poisson gap[b]		
						θ: 0-π/2	θ: 0-π	no wt.
RMSD in peak position (Hz)	1	1.0	0.720	1.690	1.516	1.803	1.309	1.688
	2	1.6	0.418	0.906	0.875	1.128	0.746	0.931
	3[c]	2.7	2.491	1.212	0.857	0.985	1.226	1.133
	4	4.3	0.163	0.324	0.334	0.445	0.282	0.364
	5	7.0	0.109	0.195	0.233	0.299	0.198	0.215
	6	11.4	0.074	0.127	0.153	0.205	0.137	0.146
	7	18.6	0.052	0.082	0.094	0.149	0.095	0.090
	8	30.2	0.048	0.053	0.062	0.108	0.068	0.058
	9	49.2	0.043	0.041	0.048	0.086	0.050	0.042
	10	80.0	0.008	0.025	0.031	0.051	0.032	0.022
RMSD in peak width (Hz)	1	1.0	0.915	1.477	1.637	3.361	2.265	1.278
	2	1.6	0.534	1.014	0.856	2.271	0.938	0.889
	3[a]	2.7	1.378	1.473	1.641	2.427	2.991	1.892
	4	4.3	0.197	0.902	0.573	1.132	0.518	0.882
	5	7.0	0.128	0.781	0.470	0.797	0.340	0.797
	6	11.4	0.075	0.610	0.329	0.584	0.216	0.634
	7	18.6	0.046	0.424	0.204	0.460	0.128	0.442
	8	30.2	0.029	0.283	0.124	0.361	0.091	0.292
	9	49.2	0.018	0.182	0.093	0.307	0.072	0.186
	10	80.0	0.011	0.120	0.126	0.290	0.070	0.127
RMSD in peak height relative to the ref. (%)	1	1.0	14.0	52.7	47.4	46.1	54.2	46.4
	2	1.6	9.1	54.8	54.1	35.1	56.0	54.0
	3[a]	2.7	5.3	46.8	28.7	10.8	17.7	49.7
	4	4.3	3.5	43.4	31.0	15.6	32.7	40.4
	5	7.0	2.1	27.6	18.2	9.3	19.4	26.7
	6	11.4	1.3	16.5	10.2	5.4	10.5	15.7
	7	18.6	0.8	9.4	5.6	3.3	5.5	9.0
	8	30.2	0.5	5.4	3.1	2.0	2.8	5.2
	9	49.2	0.5	2.7	1.6	1.5	1.6	2.7
	10	80.0	0.2	1.1	0.7	0.9	0.8	1.1
average dev. in peak height relative to the ref. (%)	1	1.0	2.5	-33.5	-37.7	-43.2	-51.2	-35.3
	2	1.6	0.3	-47.5	-48.6	-32.5	-52.8	-48.0
	3[a]	2.7	-0.3	-41.1	-23.7	1.7	-0.2	-40.5
	4	4.3	-0.2	-37.3	-26.9	-13.7	-29.2	-34.6
	5	7.0	0.1	-23.2	-15.4	-7.6	-16.7	-21.7
	6	11.4	0.1	-13.4	-8.3	-3.8	-8.9	-12.5
	7	18.6	-0.2	-7.4	-4.4	-1.8	-4.2	-6.8
	8	30.2	0.0	-4.0	-2.3	-0.6	-1.8	-3.7
	9	49.2	-0.4	-1.6	-0.9	-0.2	-0.8	-1.5
	10	80.0	0.0	-0.0	0.0	0.0	0.0	0.0
missing peaks			0.1%	11.8%	9.9%	0.6%	8.1%	10.8%

[a] Simulated fully sampled data set (t_1 = 94.8 ms) with the peak intensity scaled down by √10 and the deviations calculated from the data sets with and without the RMS noise added. This is the same input data used for Table 1, main text.

[b] Compared to the 142.2 ms (50% longer t_1) fully sampled noise free data to match the 50% extended acquisition time in the F_1 dimension of the final istHMS spectra.

[c] This peak was always placed approximately 23 Hz away from the strongest peak (#10).

NMRPipe and SMILE processing scripts used in the paper

1. peakIn.tab: peak table used to generate the simulated data used for Fig. 2

```
VARS   INDEX X_AXIS Y_AXIS XW YW HEIGHT
FORMAT %3d %9.3f %9.3f %7.3f %7.3f %+e
 0  257.0   2415.9  2.6   1.8 1.00e+05
 1  257.0   3205.9  2.6   0.7 1.63e+05
 2  257.0    489.3  2.6   0.7 2.65e+05
 3  257.0   3943.6  2.6   0.9 4.31e+05
 4  257.0   3650.6  2.6   1.0 7.01e+05
 5  257.0   1987.8  2.6   1.0 1.14e+06
 6  257.0    840.5  2.6   1.2 1.86e+06
 7  257.0   1804.2  2.6   1.2 3.02e+06
 8  257.0    823.4  2.6   1.7 4.92e+06
 9  257.0   1232.1  2.6   0.9 8.00e+06
```

2. sim2D.com: script used to simulate a 2D time domain data set using the peak table above. Note that the NMRPipe package must be updated to use the latest version of the simTimeND program for getting the correct linewidths for each peak.

```
#!/bin/csh

#-------------------------------------------------#
# To obtain the results shown, users must install   #
# Frank Delaglio's 9/1/2016 or later version of     #
# nmrPipe from the following IBBR site:             #
# <https://www.ibbr.umd.edu/nmrpipe/install.html>   #
# -scale = 1/xT/yT = 7.6293945e-6                   #
#-------------------------------------------------#

simTimeND      -in peakIn.tab  \
-xN 512        -yN 4096        \
-xT 128        -yT 1024        \
-xMODE Complex -yMODE Complex  \
-xSW  900.000  -ySW 10803.288  \
-xOBS 900.274  -yOBS 900.274   \
-xCAR 8.500    -yCAR 6.0       \
-xLAB HN       -yLAB 1H        \
-ndim 2        -aq2D States    \
-xP0 0         -yP0 0          \
-xP1 0         -yP1 0          \
-rms 13        -iseed 1548     \
-scale 7.6293945e-6            \
-out test.fid  -verb -ov
```

3. smile.com: SMILE processing script. Although the simulated time domain data in test.fid is fully sampled, there is no need to convert it to an NUS data set based on a

sampling schedule. This is because SMILE automatically sets the not-sampled points (i.e. those points not included in the sampling list) to zeros. Note that the reconstructed data has the indirect dimension already apodize and there is no need to apply a window function again before zero filling the indirect dimension.

```
#!/bin/csh

#-------------------------------------------------#
# By default, SMILE extends the acquisition time by   #
# 50%. This can be changed by setting "-xT 1024" in   #
# the SMILE command line for no extension or a        #
# different value larger than 1024 for a different    #
# amount of extension from 50%. Setting -xT < 1024 is #
# invalid and will be ignored by SMILE                #
#-------------------------------------------------#

nmrPipe -in test.fid \
| nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 0.5    \
| nmrPipe  -fn ZF -auto -zf 2                         \
| nmrPipe  -fn FT                                     \
| nmrPipe  -fn PS -p0 0 -p1 0 -di                     \
| nmrPipe  -fn EXT -x1 9ppm -xn 8ppm -sw              \
| nmrPipe  -fn TP                                     \
| nmrPipe  -fn SMILE -nDim 2 -sample nuslist          \
           -report 2 -nThread 2 -xP0 0 -xP1 0         \
| nmrPipe  -fn ZF -zf 1 -auto                         \
| nmrPipe  -fn FT                                     \
| nmrPipe  -fn PS -p0 0 -p1 0 -di                     \
| nmrPipe  -fn TP                                     \
   -verb -ov -out smile.ft2
```

4. simulation and reconstruction scripts used to generate Tables 1 and S1; visit http://spin.niddk.nih.gov/bax/software/SMILE to download the complete data sets

```
# sim2D.com, a script to generate the input time domain
# data with random noise added; setting -rms 0 yields
# a noise-free data set used as the reference for Tables 1
# and S1; note that 1536* data points (t1=142.2ms) are
# simulated, but only 10% of the first 1024* points are
# sampled for the NUS data:

#!/bin/csh

#-------------------------------------------------#
# To obtain the correct results, users must install #
# Frank Delaglio's 9/1/2016 or later version of     #
# nmrPipe from the following IBBR site:             #
# <https://www.ibbr.umd.edu/nmrpipe/install.html>   #
```

```
# -scale = 1/1280/1024 = 7.6293945e-7                #
#---------------------------------------------------#
# The peak height can be impacted by the apodization#
# function. This effect can be empiracally estimated#
# by simulating 10 peaks with the linewidth used in #
# the input peak tables and peak picking them.      #
# The intensity ratio is used to adjust the scaling #
# factor such that the S/N is fixed for the peaks   #
# across the 10 data sets simulated using different #
# random seeds and linewidth.                       #
#---------------------------------------------------#

set nTab  = 10
set seeds = ( 31875   9741  13184  26769   3866 \
              32352   1248  28452 11225    7607)
set scaF  = (7.6293945e-07 7.7102661e-07        \
             7.7934265e-07 7.9307556e-07        \
             8.0703735e-07 8.1459045e-07        \
             8.1993103e-07 8.2588195e-07        \
             8.3778381e-07 8.4472656e-07)

set n=0
while ($n < $nTab)
   @ n++
   set tabName  = `printf tab/peakIn%02d.tab $n`
   set fidName  = `printf fid/test%02d.fid   $n`
   set ftName   = `printf ft.full/test%02d.ft2 $n`

   simTimeND      -in $tabName      \
   -xN 5120       -yN 8192          \
   -xT 1280       -yT 1536          \
   -xMODE Complex -yMODE Complex    \
   -xSW  9000.000 -ySW 10803.288    \
   -xOBS 900.274  -yOBS 900.274     \
   -xCAR 5.00     -yCAR 6.0         \
   -xLAB HN       -yLAB 1H          \
   -ndim 2        -aq2D States      \
   -xP0 0         -yP0 0            \
   -xP1 0         -yP1 0            \
   -rms 5         -iseed $seeds[$n] \
   -scale $scaF[$n]                 \
   -out $fidName -verb -ov

   ft2.com $fidName $ftName

end
```

```
# ft2.com, an nmrPipe script called by sim2D.com above
# to process the fully sampled simulated time domain data:

#!/bin/csh

nmrPipe -in $1 \
| nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 0.5    \
| nmrPipe  -fn ZF -zf 2                               \
| nmrPipe  -fn FT                                     \
| nmrPipe  -fn PS -p0 0 -p1 0 -di                     \
| nmrPipe  -fn TP                                     \
| nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 1 -c 0.5    \
| nmrPipe  -fn ZF -size 8192                          \
| nmrPipe  -fn FT                                     \
| nmrPipe  -fn PS -p0 0 -p1 0 -di                     \
| nmrPipe  -fn TP                                     \
   -verb -ov -out $2



# smile.com, used to take the fully sampled time domain
# as the input, automatically sets the un-sampled
# points to zeros, and then reconstructs them:

#!/bin/csh

#-----------------------------------------------------#
# Note that nSigma is set to a slightly lower value  #
# than the default (5.0) to maximize the number of   #
# weak peaks to be reconstructed in this series of   #
# simulated data sets.                               #
#-----------------------------------------------------#

rm -fr ft.smile*

set xT=1536

set nuslist=('rdm' 'mdd' 'pg2' 'pg1' 'pg0')

set n=0

while ($n < 10)
   @ n++
   set fidName = `printf ../fid/test%02d.fid $n`
   set ftDir   = `printf ft.smile%02d $n`
   mkdir $ftDir
```

```
   set k=0
   while ($k < ${#nuslist})
      @ k++
      set listName = `printf "../nuslist/nuslist%02d.%s" \
                      $n $nuslist[$k]`
      set ftName   = `printf "%s/%sSmile.ft2"            \
                      $ftDir $nuslist[$k]`
      set logName  = `printf "%s/%sSmile.log"            \
                      $ftDir $nuslist[$k]`

      nmrPipe -in $fidName \
      | nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 0.5    \
      | nmrPipe  -fn ZF -zf 2                               \
      | nmrPipe  -fn FT                                     \
      | nmrPipe  -fn PS -p0 0 -p1 0 -di                     \
      | nmrPipe  -fn TP                                     \
      | nmrPipe  -fn SMILE -nDim 2 -sample $listName        \
                 -report 1 -nThread 2 -xT $xT -xzfSize 8192 \
                 -nSigma 4.5 -xP0 0 -xP1 0                  \
      | nmrPipe  -fn ZF -size 8192                          \
      | nmrPipe  -fn FT                                     \
      | nmrPipe  -fn PS -p0 0 -p1 0 -di                     \
      | nmrPipe  -fn TP                                     \
         -ov -out $ftName

      mv smile.log $logName

   end
end



# sparsify.py, a python program to create the NUS data sets
# from the fully sampled simulated time domain data;
# not required by SMILE because the program automatically
# sets the unsampled points to zeros.

# to be called by ist.com below

#!/usr/bin/env python

import sys, subprocess
popen = subprocess.Popen

xSize   = 1280
ySize   = 1024
hdrSize = 512

nuslist = sys.argv[1]
```

```
fullFID = sys.argv[2]
nusFID  = sys.argv[3]

inUnit = open(fullFID, 'rb')

hdrData = inUnit.read(hdrSize*4)

fidData = inUnit.read(xSize*2*ySize*2*4)

inUnit.close()

inUnit  = open(nuslist, 'r')
samples = inUnit.readlines()
inUnit.close()

outUnit = open(nusFID, 'wb')

outUnit.write(hdrData)

nSamples = 0
for i in range(len(samples)):
   thisSample = int(samples[i].strip())
   bytesStart = xSize*4*thisSample*4
   bytesEnd   = bytesStart + xSize*4*4
   fid2Write  = fidData[bytesStart:bytesEnd]
   outUnit.write(fid2Write)
   nSamples += 1

outUnit.close()

cmdArg = nusFID + ' -yT ' + str(nSamples) + ' -yN ' + \
         str(nSamples*2)
popen('sethdr ' + cmdArg, shell=True)


#ist.com, a script to reconstruct the NUS data using
istHMS:

#!/bin/csh

set yT=1536

set nuslist=('rdm' 'mdd' 'pg2' 'pg1' 'pg0')

set n=0
while ($n < 10)
   @ n++
   set fidName = `printf ../fid/test%02d.fid $n`
   set ftDir   = `printf ft.ist%02d $n`
```

```
   rm -fr $ftDir
   mkdir $ftDir

   set k=0
   while ($k < ${#nuslist})
      @ k++
      set listName = `printf "../nuslist/nuslist%02d.%s" \
                      $n $nuslist[$k]`
      set ftName   = `printf "%s/%sIST.ft2"              \
                      $ftDir $nuslist[$k]`

      rm -fr nus$n.fid nus$n.ft1 ist$n.ft1
      sparsify.py $listName $fidName nus$n.fid
      sethdr nus$n.fid -yT $yT

      nmrPipe -in nus$n.fid \
      | nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 0.5    \
      | nmrPipe  -fn ZF -zf 2                               \
      | nmrPipe  -fn FT                                     \
      | nmrPipe  -fn PS -p0 0 -p1 0 -di                     \
      | nmrPipe  -fn TP                                     \
        -out nus$n.ft1 -ov

      istHMS -ref 0 -vlist $listName -xN $yT -itr 400       \
             -user 1 -verb 0 < ./nus$n.ft1 >! ./ist$n.ft1

     sethdr ist$n.ft1 -xT 1280

     nmrPipe -in ist$n.ft1 \
     | nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 1 -c 0.5 -size $yT \
     | nmrPipe  -fn ZF -size 8192                                 \
     | nmrPipe  -fn FT                                            \
     | nmrPipe  -fn PS -p0 0 -p1 0 -di                            \
     | nmrPipe  -fn TP                                            \
       -ov -out $ftName

   end
end
```

5. expansion, conversion and reconstruction scripts for the reconstruction of the 3D (H)N(COCO)NH data. The artificial ser.nus NUS data was created by randomly picking 2480 FIDs from the fully sampled ser file

```
#!/bin/csh

#-------------------------------------------------------------#
# ser.nus is a pseudo NUS data set consisting of 2.58% of the  #
# fully sampled data in ser, selected according to the random  #
# sampling schedule in nuslist.                                #
#-------------------------------------------------------------#

nusExpand.tcl -mode bruker -sampleCount 620 -off 0 \
```

```
 -in ser.nus -out ser_full -sample nuslist

bruk2pipe -in ./ser_full \
  -bad 0.0 -aswap -DMX -decim 2856    \
  -dspfvs 20 -grpdly 67.9860382080078 \
  -xN          2048  -yN               310  -zN          310   \
  -xT          1024  -yT               155  -zT          155   \
  -xMODE        DQD  -yMODE  Echo-AntiEcho  -zMODE    Complex  \
  -xSW     7002.801  -ySW         1515.152  -zSW     1515.152  \
  -xOBS     600.433  -yOBS          60.848  -zOBS      60.848  \
  -xCAR       4.869  -yCAR         119.164  -zCAR     119.164  \
  -xLAB          HN  -yLAB            15Ny  -zLAB        15Nz  \
  -ndim           3  -aq2D          States                     \
  -out ./fid/test%03d.fid -verb -ov

rm -fr ser_full


#!/bin/csh

#-------------------------------------------------------#
# -fraction 0.55 instructs SMILE to subtract 55%        #
# (instead of 100% by default) of the reconstructed     #
# signals each iteration;                               #
# it improves the performance in the reconstruction     #
# of overlapping diagonal peaks, but slows down the     #
# reconstruction because of the increased iterations.   #
#-------------------------------------------------------#
# -maxNPks 1 restricts SMILE to pick at most one peak   #
# per 15N-15N plane in each iteration for the most      #
# accurate results, but requires more iterations.       #
#-------------------------------------------------------#
# -xNeg instructs SMILE to negate the imaginary part    #
# along the current X-axis, which reverses the          #
# spectrum in the current X-dimension;                  #
# This is recommended if -neg is required for FT in     #
# the subsequent conventional processing. However,      #
# this flag is not required for SMILE and can be        #
# ignored.                                              #
#-------------------------------------------------------#
# -maxMem 1 causes SMILE to run in the memory saving    #
# mode and limits the memory space SMILE can use to 1   #
# GB; by default (i.e. -maxMem 0), SMILE tries to       #
# allocate as much memory as it needs (~7.5 GB in       #
# this example) but aborts if the memory space is not   #
# sufficient. By setting -maxMem to the amount of       #
# memory available to the user, one may run SMILE as    #
# long as the available memory space is more than the   #
# minimum required. See Section 4 of the SMILE          #
```

```
# manual for how to estimate the smallest memory space#
# required for a particular data set.                 #
#-----------------------------------------------------#
# Note that the reconstructed data is larger than the #
# input data because of the default 50% extension     #
# from 155 to 232 points in both indirect dimensions. #
#-----------------------------------------------------#

rm -fr ft1

xyz2pipe -in fid/test%03d.fid -x                        \
| nmrPipe  -fn POLY -time                               \
| nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 1.0      \
| nmrPipe  -fn ZF -zf 2                                 \
| nmrPipe  -fn FT                                       \
| nmrPipe  -fn PS -p0 -121.2 -p1 461.3 -di              \
| nmrPipe  -fn POLY -auto -ord 2 -x1 10.5ppm -xn 6ppm   \
| nmrPipe  -fn EXT -x1 9.0ppm -xn 7.6ppm -sw -round 2   \
| pipe2xyz -out ft1/test%03d.ft1 -z

rm -fr ft1.smile

xyz2pipe -in ft1/test%03d.ft1 -x                        \
| nmrPipe  -fn SMILE -nDim 3 -nThread 4                 \
           -sample nuslist -fraction 0.55 -report 2     \
           -maxIter 500 -maxNPks 1 -maxMem 1            \
           -xP0 0 -xP1 0 -xNeg                          \
           -yP0 0 -yP1 0                                \
| pipe2xyz -out ft1.smile/test%03d.ft1 -x

rm -fr ft.smile

xyz2pipe -in ft1.smile/test%03d.ft1 -x                  \
| nmrPipe  -fn ZF -zf 2                                 \
| nmrPipe  -fn FT -neg                                  \
| nmrPipe  -fn PS -p0 0.0 -p1 0.0 -di                   \
| nmrPipe  -fn TP                                       \
| nmrPipe  -fn ZF -zf 2                                 \
| nmrPipe  -fn FT                                       \
| nmrPipe  -fn PS -p0 0.0 -p1 0.0 -di                   \
| pipe2xyz -out ft/test%03d.ft3 -y

rm -fr *.dat

proj3D.tcl -in ft/test%03d.ft3
```

6. expansion, conversion and reconstruction scripts for the reconstruction for the 4D methyl HMQC-NOESY-HMQC data

```
#!/bin/csh

#Z=F1, 13C indirect dim (Cnoe)
#Y=F2, 1H indirect dim  (Hnoe)
#A=F3, 13C indirect dim (Cm)
#X=F4, 1H direct dim    (Hm)

rm -fr ser_full

nusExpand.tcl -mode bruker -sampleCount 5600 -off 0 \
 -in ./ser -out ./ser_full -sample ./nuslist

rm -fr fid

bruk2pipe -in ./ser_full \
  -bad 0.0 -aswap -AMX -decim 2496     \
  -dspfvs 20 -grpdly 67.9842376708984  \
  -xN        2048  -yN        112  -zN        160  -aN        160  \
  -xT        1024  -yT         56  -zT         80  -aT         80  \
  -xMODE      DQD  -yMODE Complex  -zMODE Complex  -aMODE Complex  \
  -xSW   8012.821  -ySW  1201.923  -zSW  1984.127  -aSW   984.127  \
  -xOBS   600.433  -yOBS  600.433  -zOBS  150.981  -aOBS  150.981  \
  -xCAR     4.897  -yCAR    0.794  -zCAR   20.391  -aCAR   20.391  \
  -xLAB        Hm  -yLAB     Hnoe  -zLAB     Cnoe  -aLAB       Cm  \
  -ndim         4  -aq2D   States                                   \
| pipe2xyz -x -out ./fid/test%03d.fid -verb -ov -to 0

rm -fr ser_full


#!/bin/csh

#-------------------------------------------------------#
# The 3 indirect dimensions are extended by 50% from    #
# 56, 80, and 80 (corresponding to -yT, -zT, and -aT    #
# in fid.com) to 84, 120, and 120, respectively;        #
# see the input and reconstructed data sizes by:        #
# showhdr ft1/test001.ft1 ft1.smile/test001.ft1         #
# One can override this SMILE default by specifying     #
# -xT 56 -yT 80 -zT 80 to keep the size of data matrix  #
# unchanged during the reconstruction, or to a larger   #
# number than the corresponding -T in fid.com to make   #
# a different amount of extension. A number smaller     #
# than the corresponding -T value in fid.com (for       #
# example -xT 50 smaller than -yT 56 in fid.com) is     #
# invalid and therefore is always ignored by SMILE.     #
# Note that the x-axis in fid.com is always flipped to  #
```

```
# the a-axis, and the y-, z-, and a-axes in fid.com    #
# become x-, y-, and z-axes, respectively, for SMILE.  #
#------------------------------------------------------#
# "-maxMem 40" makes SMILE to run in the chunk-wise    #
# memory saving mode. Without this, by default, SMILE  #
# tries to allocate more than 1.2 TBs of memory for FT,#
# which may exceed the memory space one has access to. #
# However, when the maximal amount of memory is limited#
# to 40 GB, SMILE can still run even if one does not   #
# have 1.2 TB memory, as long as the memory space is   #
# sufficient for FT of at least 3 cubes of the 4D data.#
#------------------------------------------------------#

#X=F4, 1H direct dim
#Y=F2, 1H indirect dim
#Z=F1, 13C indirect dim (Cnoe)
#A=F3, 13C indirect dim (Cm)

rm -fr ft1

xyz2pipe -in fid/test%03d.fid -x                          \
| nmrPipe  -fn POLY -time                                  \
| nmrPipe  -fn SP -off 0.5 -end 0.98 -pow 2 -c 0.5        \
| nmrPipe  -fn ZF -zf 2 -auto                              \
| nmrPipe  -fn FT                                         \
| nmrPipe  -fn POLY -auto -ord 2 -x1 3ppm -xn -1ppm       \
| nmrPipe  -fn EXT -x1 1.4ppm -xn -0.7ppm -sw -round 2    \
| nmrPipe  -fn PS -p0 122 -p1 0 -di                       \
| pipe2xyz -ov -out ft1/test%03d.ft1 -a

rm -fr ft1.smile

date

xyz2pipe -in ft1/test%03d.ft1 -x                        \
| nmrPipe  -fn SMILE -nDim 4 -thresh 0.9 -nThread 16     \
           -maxMem 40 -sample nuslist -report 2         \
           -xP0   0 -xP1   0 -xNeg                      \
           -yP0   3 -yP1   0 -yNeg                      \
           -zP0  87 -zP1 180                            \
| pipe2xyz -out ft1.smile/test%03d.ft1 -x

rm -fr ft

xyz2pipe -in ft1.smile/test%03d.ft1 -x                   \
| nmrPipe  -fn ZF -zf 2 -auto                            \
| nmrPipe  -fn FT -neg                                    \
| nmrPipe  -fn PS -p0 0 -p1 0 -di                         \
```

```
| nmrPipe  -fn TP                                          \
| nmrPipe  -fn ZF -zf 2 -auto                              \
| nmrPipe  -fn FT -neg                                     \
| nmrPipe  -fn PS -p0 3 -p1 0 -di                          \
| pipe2xyz -out ft/test%03d.ft3 -y

xyz2pipe -in ft/test%03d.ft3 -z                           \
| nmrPipe  -fn ZF -zf 2 -auto                              \
| nmrPipe  -fn FT                                          \
| nmrPipe  -fn PS -p0 87 -p1 180 -di                       \
| pipe2xyz -out ft/test%03d.ft4 -z

date

rm -fr dat ft/test.ft3

proj4D.tcl -in ft/test%03d.ft4
```

References:

Coggins, BE, Werner-Allen, JW, Yan, A, Zhou, P (2012) Rapid Protein Global Fold Determination Using Ultrasparse Sampling, High-Dynamic Range Artifact Suppression, and Time-Shared NOESY. J Am Chem Soc 134: 18619-18630.

Goddard, TD, Kneller, DG (2008). Sparky 3. University of California, San Francisco.

Hyberts, SG, Milbradt, AG, Wagner, AB, Arthanari, H, Wagner, G (2012) Application of iterative soft thresholding for fast reconstruction of NMR data non-uniformly sampled with multidimensional Poisson Gap scheduling. J Biomol NMR 52: 315-327.

Kazimierczuk, K, Orekhov, VY (2011) Accelerated NMR Spectroscopy by Using Compressed Sensing. Angewandte Chemie-International Edition 50: 5556-5559.

Stanek, J, Augustyniak, R, Kozminski, W (2012) Suppression of sampling artefacts in high-resolution four-dimensional NMR spectra using signal separation algorithm. J Magn Reson 214: 91-102.