FDM Processing Guide

Justin Douglas (justindo@gmail.com)

Part 1. The mechanics of FDM-processing.

The basic concept

The basic idea of FDM processing is that it is a series of format conversions. The spectrometer specific data is converted into nmrPipe format using the nmrPipe software package (Delaglio et. al). Then the nmrPipe format file is converted into an ASCII file by the program pipe3D.M for 3D data or pipe4D.M for 4D data. This results in a large ASCII file, which I typically call FDM_input.txt. FDM_input.txt and a parameter files that I call in_3dct are then read into the FDM processing program, called fdm2k_3dct_3Dbin for 3D data or fdm2k_4dct_4Dbin for 4D data. The output is a binary file called sp3d.FDM_CT23. Finally, the program fort2pipe converts this file back into an nmrPipe format for subsequent analysis. The purpose of this FDM Processing Guide is to help people new to FDM start to process their data.

Step 1. Convert spectrometer-specific data into nmrPipe format.

The first step is to convert the spectrometer specific data files into nmrPipe format. This step is required for conventional data processing with nmrPipe and should be familiar to most users. See the man pages for var2pipe for additional assistance by typing the following in the command prompt:

```
%man var2pipe
```

A typical var2pipe script for conversion of a 3D hCCH-COSY of RNA is shown below:

```
var2pipe -in ./fid -noaswap -aqORD 1  \
  -xN              2048  -yN                16  -zN                16  \
  -xT              1024  -yT                 8  -zT                 8  \
  -xMODE        Complex  -yMODE         Complex  -zMODE         Complex  \
  -xSW         6000.600  -ySW          4000.000  -zSW          2500.000  \
  -xOBS         499.597  -yOBS          125.698  -zOBS          125.698  \
  -xCAR           4.794  -yCAR          100.035  -zCAR          141.423  \
  -xLAB              H6  -yLAB               C5  -zLAB               C6  \
  -ndim               3  -aq2D           States                          \
  -out ./data/test%03d.fid -verb -ov
```

The output is a directory called data/.

Step 2. Conversion of nmrPipe data format to file format for FDM-processing.

Unfortunately, the FDM software cannot use the nmrPipe format files directly. The data must first be converted using the program pipe3D.M for 3D data or pipe4D.M for 4D data. These are scripts written in the Mscript Macro Language (see https://helpdesk.islandnet.com/mscript/guide.php#introduction for more information on Mscript). To execute these scripts type the following in the command prompt.

```
%M pipe3D.M -str inName data/test%03.0f.fid outName FDM_input.txt -quit
```

Step 3. FDM calculation

At this point we are ready to do an FDM calculation. To start the program type the following in the command prompt

```
%fdm2k_3dct_3Dbin_v3.1 < in_3dct
```

This invokes the executable file “fdm2k_3dct_3dbin_v3.1” with input from the parameter file in_3dct. The file in_3dct contains all of the FDM-processing parameters and will be discussed in depth in the 2nd part of this guide. A sample parameter file is shown below:

```
'FDM_input.txt', 0                    ! signal, theta
 256,   8,   8                 ! Nsig1,Nsig2,Nsig3
 1000 2000                             ! wmin1 wmax1
 -2001 2001                            ! wmin2 wmax2
 -1251 1251                            ! wmin3 wmax3
 'FDM_CT23'                            ! method string
 16,  6,  6, 1.1, 1.0, 1.0             ! Nb1,Nb2,Nb3
 'sp3d' 1                      ! 3D spectral file
 'none', 'none', 'none'  1     ! 1D projection files
 'none', 'none', 'none'  1     ! 2D projection files
  512   128     64                      ! Nsp1-3 (digital resolutions)
 5   45  45                           ! Gamm1-3 (smoothing threshold)
 'G',   1.0                            ! Lineshape, cheat
 0.0001                                ! q**2 (regularization level)
```

Step 4. Convert output of FDM calculation to nmrPipe format file.

The final step is to convert the output of the FDM calculation into nmrPipe format file. This step is done using the program fort2pipe, which was designed to mimic the var2pipe module of nmrPipe. A typical fort2pipe script is shown below:

```
fort2pipe -in sp3d.FDM_CT23 -ndim 3   \
  -xN          512  -yN         128  -zN          64 \
  -xT          512  -yT         128  -zT          64 \
  -xOBS    499.597  -yOBS   125.698  -zOBS   125.698 \
  -xCAR      4.794  -yCAR   100.035  -zCAR   141.423 \
  -xLAB         H6  -yLAB        C5  -zLAB        C6 \
  -out fdmdata/test%03d.dat
```

Part 2. Optimizing FDM-processing parameters.

One of the challenges in using FDM is optimizing the FDM-processing parameters. This section will define the relevant parameters and give examples of how these parameters impact the resulting spectra. This information should help beginners choose reasonable processing parameter.

The relevant processing parameters are:

theta – zero order phase correction in the directly acquired dimension.

Nsig – number of time-domain data points to be included in the calculation

wmin and wmax – the spectral region to be processed (in Hz from the carrier; note that positive numbers are downfield)

Method String – sets options for CT doubling.

Nb – number of basis functions per window.

Nsp – digital resolution of reconstructed spectrum (must equal values in fort2pipe script)

gamma – linewidth parameter

q^2 – regularization parameter

The remainder of this document contains examples for how each FDM-processing parameter affects the spectrum. I used a base-specific 3D hCCH-COSY recorded on IRE to test these processing parameters. The table below denotes the optimal processing parameters. On the following pages, a 2D H6-C6 projection is shown for the data processed by varying only one parameter. These results should be helpful for optimizing processing parameters. The major conclusion that can be drawn by inspecting these projection is that, initially only q2, g in the indirect dimensions, Nsig and the phases. Other parameters have a more modest effect on the spectrum.

Table 1: Optimized FDM-processing parameters.

Nsig	w_{min}, w_{max}(F3)	w_{min}, w_{max}(F2)	w_{min}, w_{max}(F1)	Nb1	Nb2	Nb3	γ1	γ2	γ3	q^2
256	1000, 2000	-2000, 2000	-1250, 1250	16	6	6	5	45	45	1e-4

Figure 1. The effects of the regularization parameter on the 2D H6-C6 projection of the base-specific 3D hCCH-COSY on the IRE RNA.

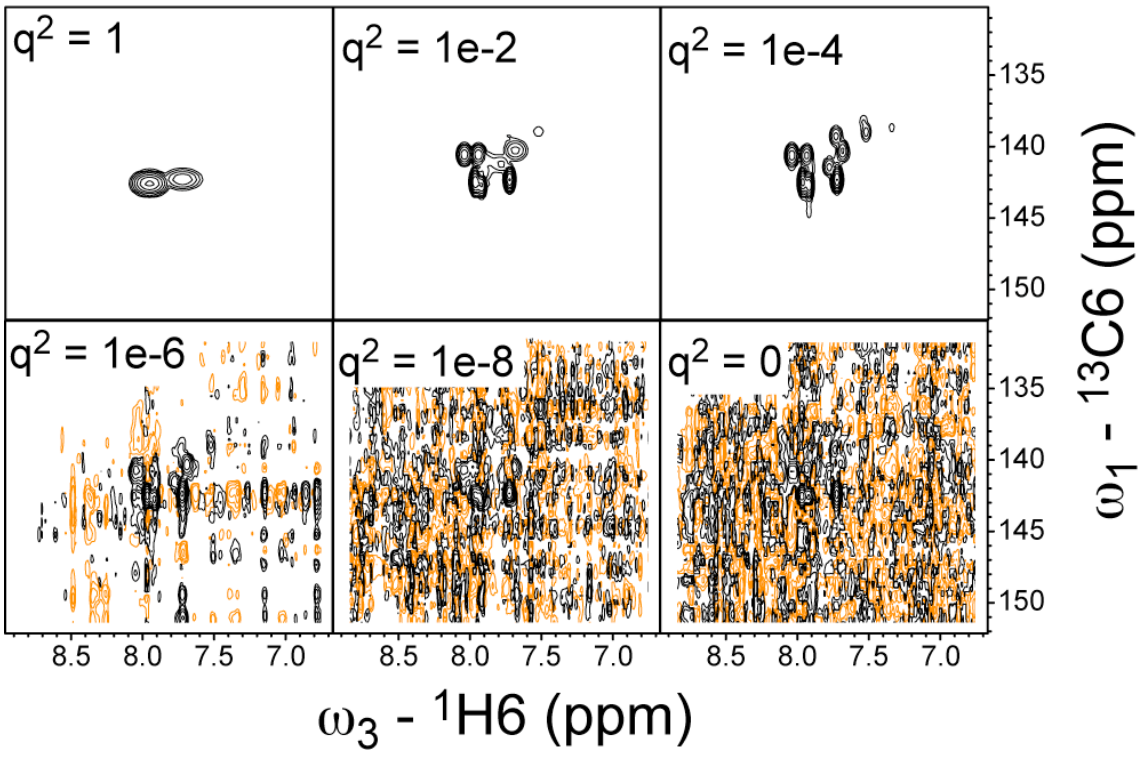


Figure 2. The effects of the linewidth parameter in the indirect dimensions (g2 for the C5 dimension and g3 for C6 dimension) on the 2D H6-C6 projection of the base-specific 3D hCCH-COSY on the IRE RNA.

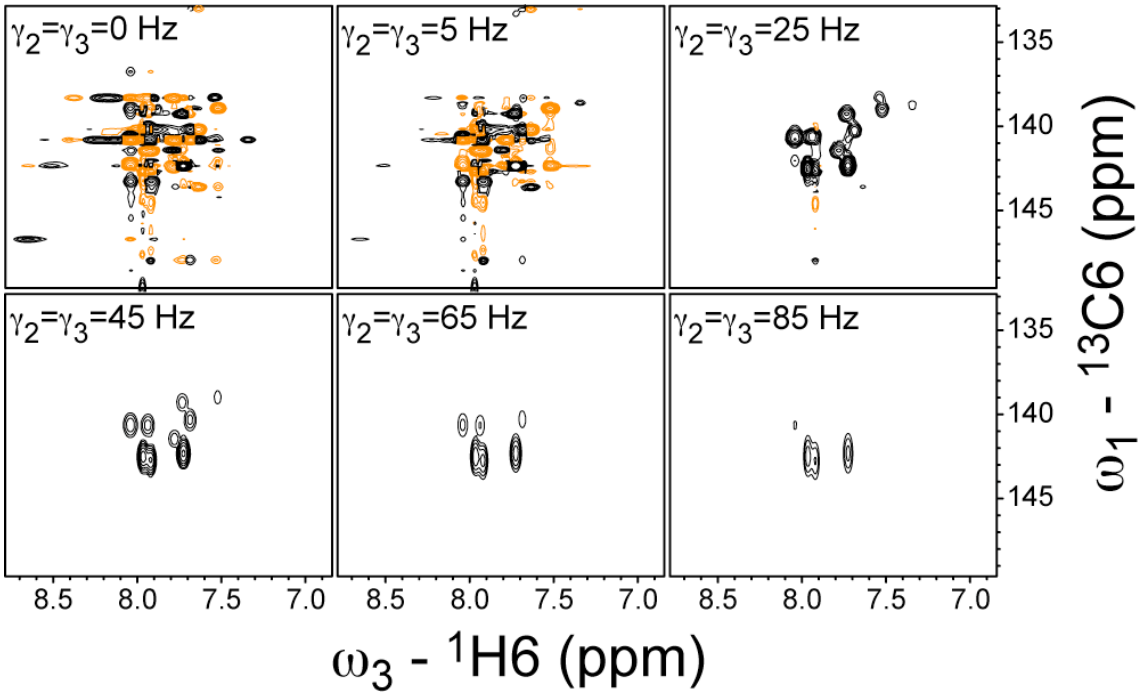


Figure 3. The effects of the phase in the C6 dimension on the 2D H6-C6 projection of the base-specific 3D hCCH-COSY on the IRE RNA.

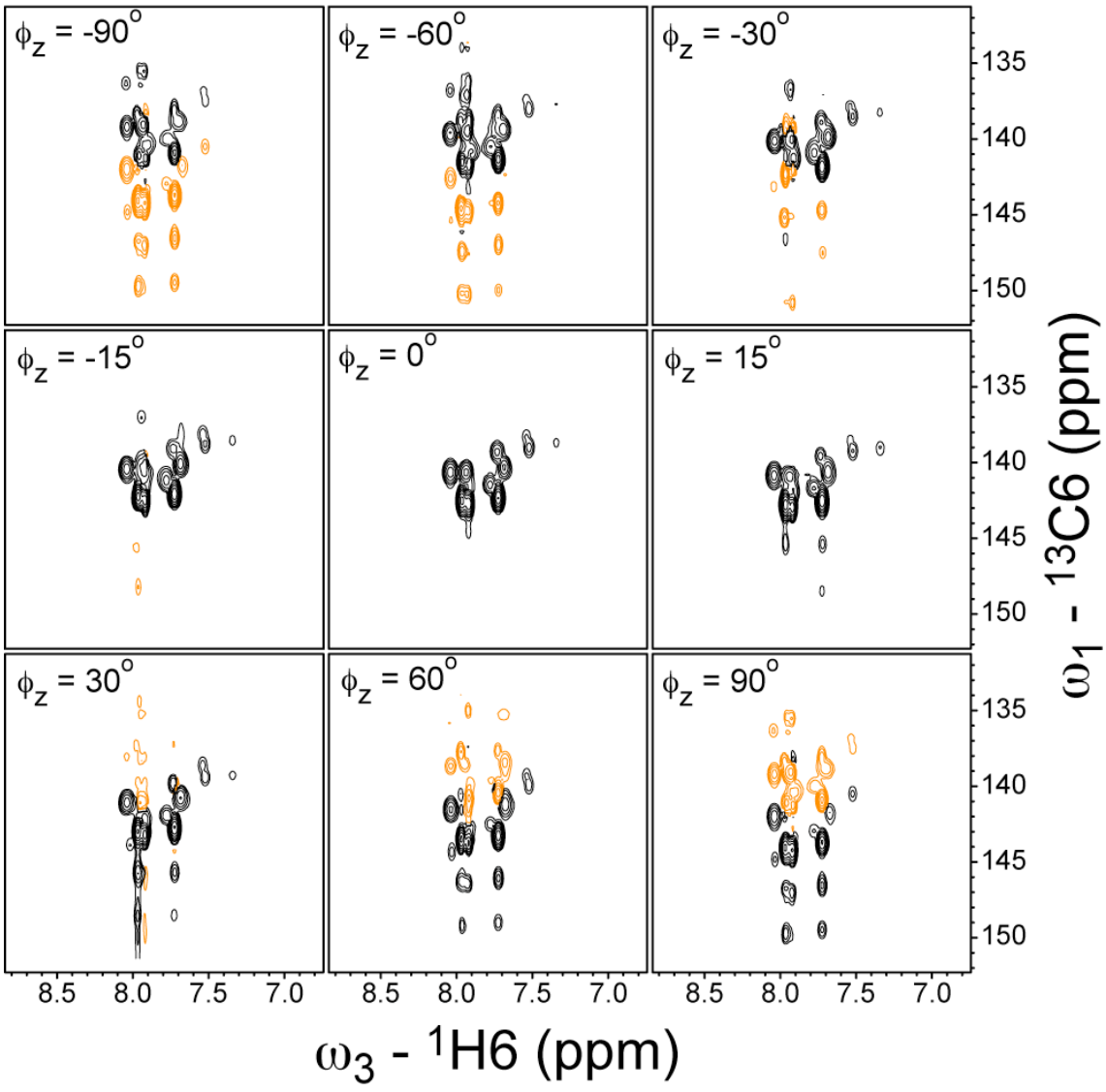


Figure 4. The effects of the number of data points in the directly acquired dimension used in the FDM calculation on the 2D H6-C6 projection of the base-specific 3D hCCH-COSY on the IRE RNA.

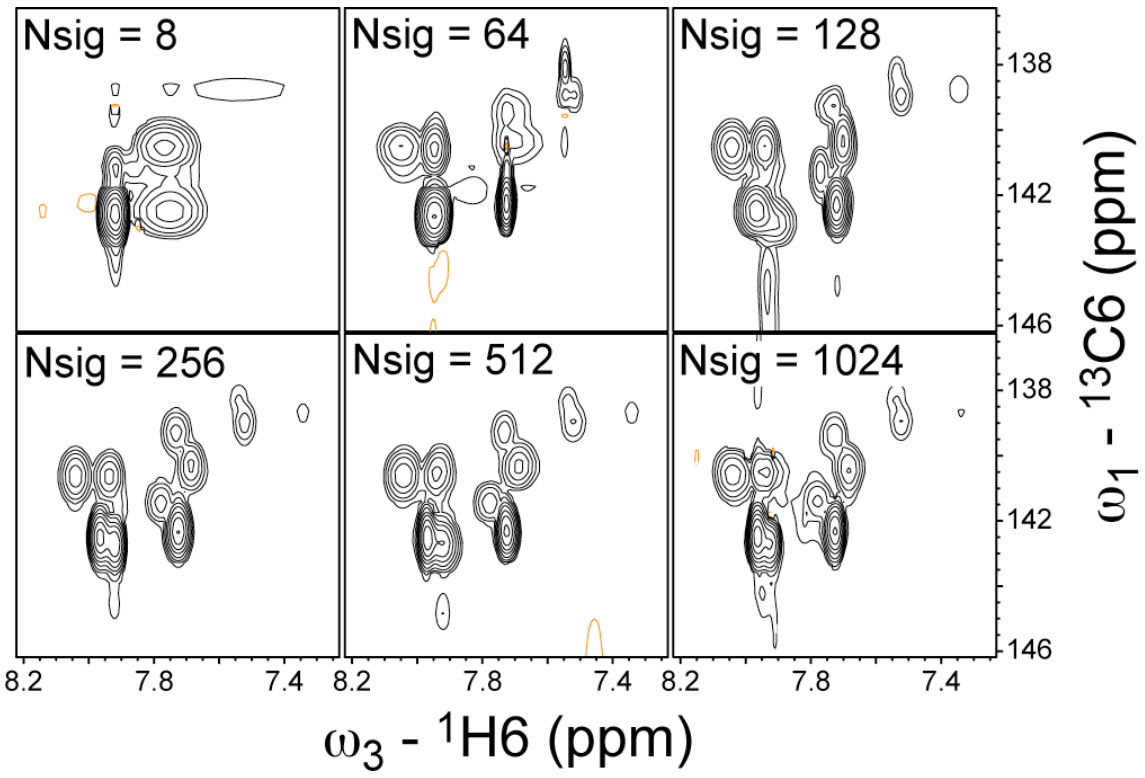


Figure 5. The effects of the linewidth parameter in the directly acquired dimension on the 2D H6-C6 projection of the base-specific 3D hCCH-COSY on the IRE RNA.

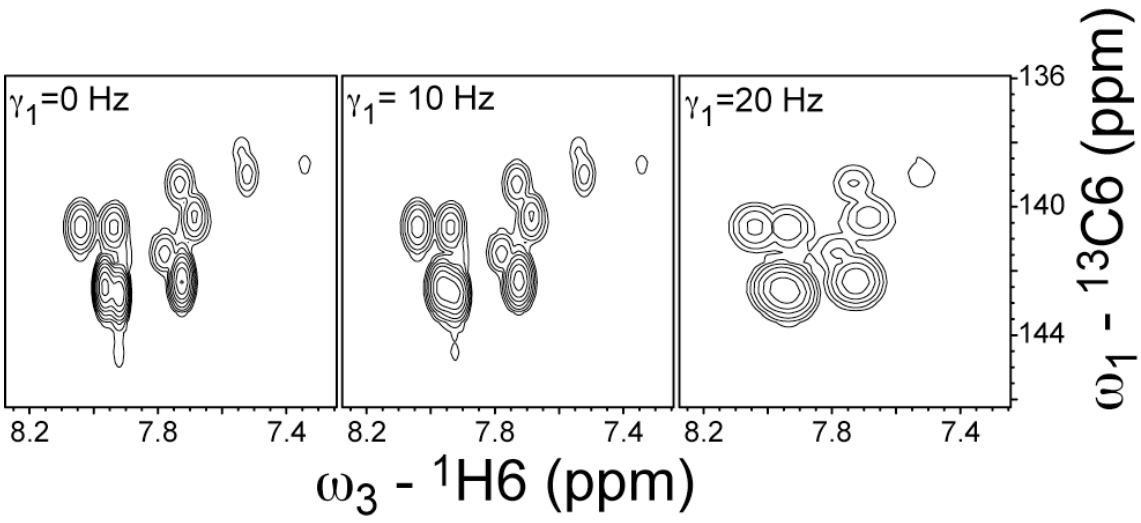


Figure 6. The effects of the number of basis function per window in the directly acquired dimension on the 2D H6-C6 projection of the base-specific 3D hCCH-COSY on the IRE RNA.

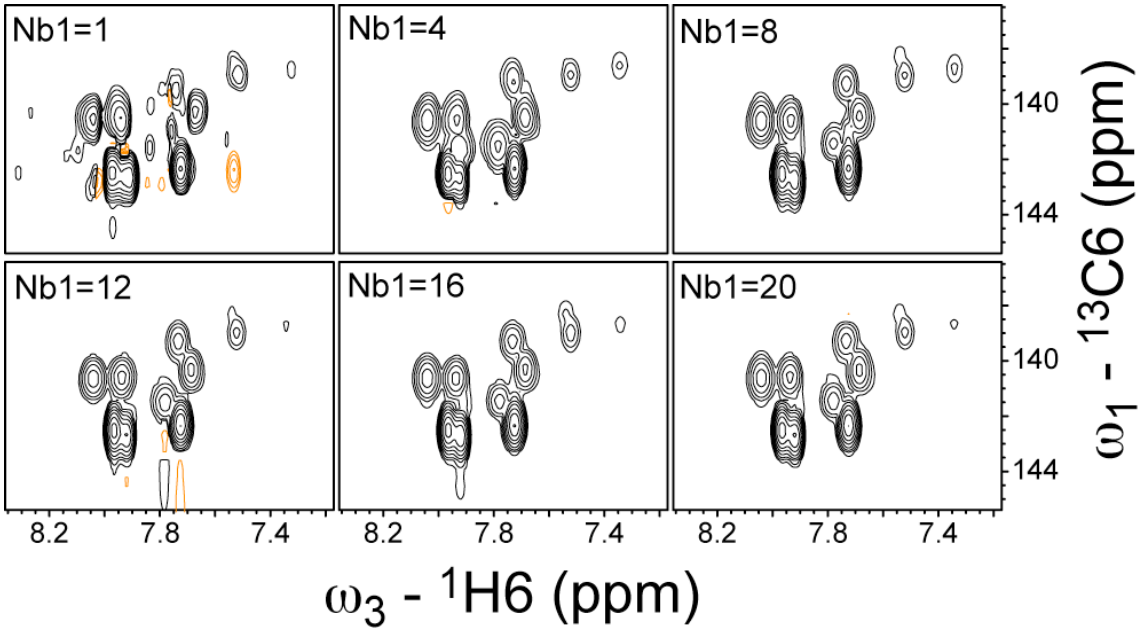
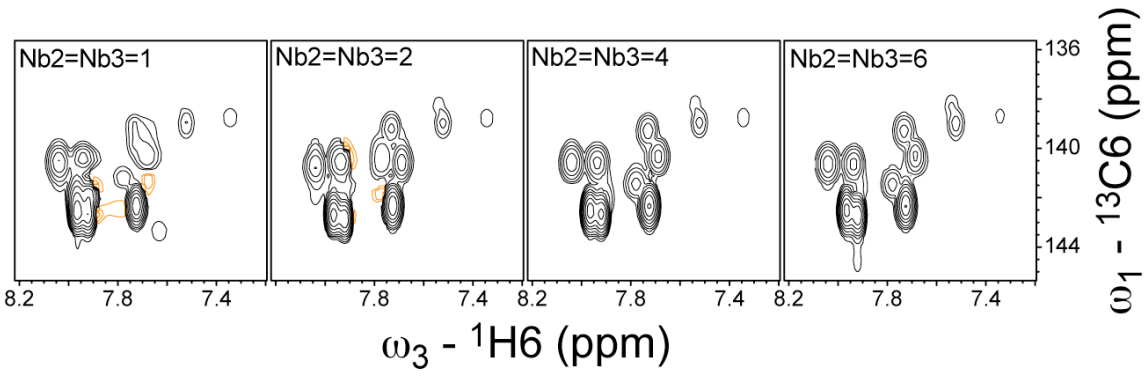
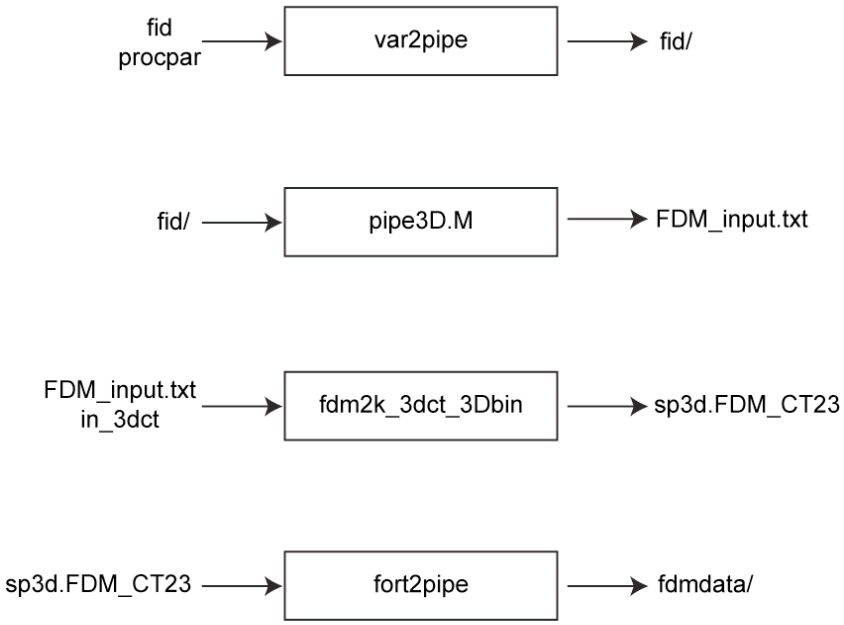


Figure 7. The effects of the number of basis function per window in the indirectly acquired dimensions on the 2D H6-C6 projection of the base-specific 3D hCCH-COSY on the IRE RNA.



Flow Chart for FDM-processing



Justin T. Douglas
justindo@gmail.com