

Using NOAH Supersequences

Quick set-up

NOAH experiments are set-up most easily from the predefined parameter sets, thus:

- Install NOAH parameter sets, pulse program(s) and processing AU programs
- Execute *paracon*, load NOAH parameter set, load pulse values (*getprosol*)

Full experimental setup

The NOAH experiments are setup in a manner similar to conventional 2D heteronuclear ^{13}C experiments, but with the following specific requirements:

- The number of memory blocks used for data acquisition must be defined with the acquisition parameter *NBL*. This must be set equal to the number of NOAH modules in the sequence e.g. *NBL=4* for a NOAH-4 experiment.

NBL	4	Number of blocks (of acquisition memory)
-----	---	--
- The TD1 data size must be increased (scaled) according to the number of NOAH modules in the sequence. Thus, a NOAH-4 requiring 256 t_1 points for each NOAH module dictates that TD1 be set to $256 \times 4 = 1024$ points.
- Since only a single F1 frequency domain is explicitly defined (typically for the ^{13}C dimension) the scaling constants for the F1 spectral widths for all other F1 dimensions (^{15}N and/or ^1H) must be set for the correct definition of these frequencies in the final data sets.
 - *cnst10* defines the $^{13}\text{C}:^1\text{H}$ spectral width (as Hz) ratio- **this is calculated within the pulse programs**. Set the value *CNST10=1* (default) in the parameters for correct processing.
 - *cnst20* defines the $^{13}\text{C}:^{15}\text{N}$ spectral width (as Hz) ratio (if ^{15}N experiments are used)- **this must be defined manually as *CNST20***.

All required data processing is performed by executing a single AU script (*splitx_au*) either manually or via XAUP.



Overall, the process makes use of the following AU processing scripts:

- **splitx_au**: splits the single NOAH data set into *NBL* individual data sets for processing as separate experiments. For a NOAH experiment acquired in EXPNO *X*, **splitx_au** will produce data sets numbered *X001*, *X002*, *X003* ...etc according to the number of NOAH modules.
- **noah_xxxx**: this is a collection of AU scripts that process the split, individual experiments which are called by the **splitx_au** program. They must be set as the user-defined processing parameters *USERP1...USERPN* (for *N* NOAH modules) according to the modules used. Arguments after the AU program file name will define the quadrature detection mode employed e.g. *noah_tocsy States* (default is echo-antiecho if nothing defined; see comments in pulse programs).

User parameters		
USERP1	noah_hmbc	User processing par. 1
USERP2	noah_hsqc	User processing par. 2
USERP3	noah_clip-cosy	User processing par. 3
USERP4		User processing par. 4

- **proc_noah**: this is called by the *USERP* processing programs to perform the appropriate FT operations for conventional or NUS data sets.
- **fixF1**: corrects the F1 chemical shift axis for ^1H (i.e. homonuclear 2D experiments). Uses *cnst10* as input (defined by pulse program).
- **fixF1n**: corrects the F1 chemical shift axis for ^{15}N . Uses *cnst20* as input (defined manually).

NOAH under automation

NOAH parameter sets may be used directly under automation within ICONNMR, as for conventional experiment parameter sets. Data acquisition is initiated by the Bruker standard AU program `au_zgcosy` ($AUNM = au_zgcosy$). Following this, processing is handled via the `splitx_au` AU program which must be defined in the processing parameters ($AUNMP = splitx_au$), together with the user defined AU programs `USERP1`, `USERP2`... `USERPN` (see above).

For the optional implementation of *Wavemaker* statements for shaped pulse definition under automation the line:

```
XCMD ("wvm -q")
```

should be included in the acquisition AU program, prior to the ZG statement.

Implementation of NUS under automation can be achieved by running the python NUS set-up script `noah_nus.py` in real time by including the line:

```
XPY ("noah_nus")
```

in the acquisition AU program, prior to the ZG statement.

Non-Uniform Sampling (NUS) in NOAH

The NOAH supersequences are written to be compatible with NUS. The homonuclear modules of the NOAH sequences employ either echo-antiecho or States F1 quadrature detection for NUS compatibility.

Currently the standard Bruker NUS implementation in TOPSPIN cannot be used directly with the supersequences and NUS sampling must be defined explicitly within the sequence. A python script (`noah_nus.py`) is available that will convert a NOAH experiment into the NUS variant and will define the necessary parameters for executing NUS. The modified pulse program is stored with a `.nus` extension and may be used for future NUS data collection. The sampling itself is defined within the pulse program using the following definitions for $d0$ (heteronuclear t_1) and $d10$ (homonuclear t_1) periods.

```
define list<loopcounter> t1list=<${VCLIST}> ;define NUS list
"d0=in0*t1list+3u" ; calculate t1 intervals explicitly
"d10=in10*t1list+3u"
lm t1list.inc ;increment t1 list pointer on looping
```

The `VCLIST` (named "noah") contains the sampling intervals and is generated automatically by the python script according to the requested `NusAMOUNT`.

The user therefore follows the following simple protocol to set up the NUS experiment:

1. Load the appropriate NOAH experiment parameter set
2. Define the % NUS sampling required in the usual manner (`NusAMOUNT [%]`)
3. Execute the python script `noah_nus.py`.

The `FnTYPE` **MUST** be left as *traditional(planes)*- **DO NOT** change this to *non-uniform_sampling*.

The acquired NOAH data set may be processed using the standard NOAH processing AU scripts which will recognise the use of NUS sampling and will process data accordingly. A Hilbert transform is also called to allow 2D phasing.