The below contains some useful information on acquiring and processing NOAH experiments, especially the new ^{15}N modules provided in the current package.

Acquiring a NOAH experiment: an overview

Nearly all of the setup is the same as for a standard ^{13}C–^{1}H 2D experiment. There are two parameters which require additional consideration:

- **NBL** should be set to the total number of modules, e.g. 4 for a MSCN supersequence. (Set this either in `ased`, or by typing `nbl` into the TopSpin command line.)
- **TD1** (i.e. TD in the F1 dimension) must be its original value multiplied by the total number of modules. For example, if you are running a supersequence with 4 modules and want to have 256 t_1 increments in each module, TD1 must be set to 256 × 4 = 1024.

All other parameters are fully documented in the pulse programmes, and nearly all of these are the same as in the standard experiments. There are a few parameters which are NOAH-specific:

- **cnst16**: factor to lengthen CTP gradients in ^{15}N spectra by. We recommend setting this such that cnst16*p16 is around 2–2.5 ms. In other words, if your "standard" gradient duration p16 is 1 ms, set cnst16 = 2–2.5. This helps to suppress artefacts in the spectra which arise from imperfect pulses.
- **cnst32**: in supersequences with multiple ^{13}C spectra (e.g. HSQC-TOCSY + HSQC), cnst32 tells you how much of the one-bond C–H magnetisation to allocate to the first of the two, where 1 means "all of it" and 0 means "none". The remainder goes to the second module. The optimum amount to use depends on what these two modules are, as well as other parameters e.g. acquisition times and relaxation rates, but something between 0.7 and 0.9 usually works well.
- **cnst39**: *k*-scaling factor for ^{15}N modules. Set this to a value larger than 1 to reduce TD1 for the ^{15}N experiment by a factor of cnst39; in return NS will be increased by a factor of cnst39. The impact is not huge, but sometimes sensitivity gains can be attained.
- **cnst40**: for ^{15}N experiments this is the nitrogen SW in ppm. (The nitrogen transmitter offset is O3P.)

What are acquisition flags, and how do I use them?

Acquisition flags are ways of conditionally running (technically, compiling) parts of a pulse programme. They are useful when we don't want to create two versions of what's essentially the same module with just small changes. In this case, you can enable acquisition flags to turn on/off optional features. The available acquisition flags are listed near the top of the pulse programme. To turn on an acquisition flag, such as `-DEDIT`, simply include it in the TopSpin acquisition parameter **zgoptns** (accessible via `ased`).

There are currently a handful of acquisition flags:

- **-DEDIT** and **-DEDIT1**: enables multiplicity editing in HSQC spectra.
- **-DTEDIT**: enables multiplicity editing in HSQC-TOCSY spectra.
- **-DINVERT** and **-DINVERT1**: in HSQC-TOCSY spectra, causes 'direct' (HSQC) and 'indirect' (HSQC-TOCSY) responses to have opposite signs.

To see whether you need the flags with '1' in them, please check the header of the pulse programme, which will always describe these in detail. Usually, the flags ending in '1' are applied to the first of two ^{13}C modules in a supersequence (e.g. with HSQC-TOCSY plus HSQC, then -DINVERT1 refers to the HSQC-TOCSY and -DEDIT refers to the HSQC). If you only have one ^{13}C module, it's unlikely you will ever need the '1'-suffixed flags.

There is also the -DNUS flag, which (partially) turns on non-uniform sampling, but you should not set that manually unless you're absolutely certain what you're doing. Instead, please use the `noah_nus.py` script. See: *“Can non-uniform sampling be used with NOAH sequences?”* for more information.

Does rga work as expected?

No, it doesn't. It tries to optimise the receiver gain for the first module, instead of the entire supersequence as a whole. If the first module has low intrinsic sensitivity (which is typically the case), then `rga` will suggest unreasonably large values for the receiver gain, leading to ADC overflows for the more sensitive modules.

If you want to use `rga`, you will need to run it with a different pulse programme which corresponds to the most sensitive module. For example, for a MSCN supersequence, you should run `rga` with the pulse programme set to a COSY.

Alternatively, instead of simply acquiring with `zg`, you can run `au_zgcosy`. This performs `rga` on a pulse-acquire sequence before acquiring the actual data.

During acquisition, is it normal that TopSpin shows weird numbers in the progress bar?

Yes. Using the same example as before, if you are running a NOAH-4 experiment with 256 t_1 increments (i.e. TD1 = 1024), then the progress line in the status bar will inch towards "1024/256".

Is it possible to have different values of SW/TD2/RG/<other parameter> for different modules?

In a way, yes, but as far as we are aware it has to be hardcoded in the pulse programme (and often in a nontrivial way). That would make the pulse programme overly specialised for a particular sample/spectrometer, and isn't something we want to do here.

Processing a NOAH experiment: an overview

1. Install the AU and Python scripts by copying them to the appropriate directory in TopSpin.
2. Process the data by entering `splitx_au` into the TopSpin command line. This will create new datasets with expnos beginning from (1000 × *m*) + 1, where *m* is the expno of the original NOAH data.

If you are using old versions of the pulse programme, there is an additional step where you need to specify the userP1 through userP5 parameters. For example, in a BSC experiment, you would specify (userP1, userP2, userP3) = (noah_hmbc, noah_hsqc, noah_cosy) respectively. This is not necessary if you are using the AU scripts and pulse programmes provided in this package. Please do get in touch if you need extra information on this.

I'm using macOS Catalina / Big Sur and the processing doesn't work.

On Catalina: Please update TopSpin to version 4.1.0 or later. On older versions of TopSpin for macOS, there is a bug in TopSpin which causes certain parts of AU programmes to not work on Catalina.

On Big Sur: AU programmes are broken even on TopSpin 4.1.1. However, there is a fix available for 4.1.1, courtesy of Gareth Morris and Marshall Smith (University of Manchester).

1. Install the XCode Command Line Tools by running `xcode-select --install` in a terminal.
2. In the file `/opt/topspin4.1.1/exp/stan/nmr/au/makeau`: uncomment line 20 (by removing the `#`), and remove the text `-Wl,-lcrt1.o` from line 494.
3. In `/opt/topspin4.1.1/prog/include/lib/libcb.h`: change line 28 from `#include<values.h>` to `#include<limits.h>`, then below that, add two new lines `#define MININT INT_MIN` and `#define MAXINT INT_MAX`.
4. In `/opt/topspin4.1.1/prog/include/lib/uni.h`: comment out line 182 by adding a `#` at its beginning.

Can non-uniform sampling be used with NOAH sequences?

NUS can be enabled in the pulse sequence, but **not** in the conventional way of setting FnTYPE to 'non-uniform sampling'. For all pulse programmes downloaded from this website, please follow the steps below:

1. Install the noah_nus2.py script by copying it to the appropriate TopSpin folder.
2. Set up your experiment normally without NUS first, with the full value for TD1. Don't change any other parameters, such as FnTYPE.
3. Specify the parameter NUSAmount and then run `noah_nus2` in TopSpin.

To disable NUS on a dataset where it was previously enabled, run `noah_nus2 off`.

The optimal sampling density depends on what spectra you're trying to run as well as your sample. However, there is nothing special about NOAH in this regard: the typical considerations (e.g. how sparse the spectrum is) are the same, and whatever works for the individual spectra will also work on NOAH.

Incompatibilities: The script will not work on old versions of pulse sequences, e.g. those found in the Supporting Information of older papers. It is also not compatible with the cnst39 scaling in ^{15}N modules respectively. It also cannot be used with QF COSY. The behaviour of `noah_nus2.py` is currently as follows:

- If you try to enable NUS on a ^{1}H module requiring scaling (i.e. using cnst37), or a QF COSY, it will simply refuse to work.
- If you try to enable NUS on a ^{15}N experiment with cnst39 > 1, cnst39 will be changed back to 1.

How can NOAH experiments be set up for use in automation?

The acquisition AU programme (AUNM) should be set to `au_zgcosy`, to avoid issues with `rga` (see above). The processing AU programme (AUNMP) should be set to `splitx_au`. If the pulse programmes and AU scripts are up-to-date, then there is no need to set the user processing parameters.

WaveMaker shaped pulses can optionally be generated before acquisition by adding the statement `XCMD("wvm -q")` into the acquisition AU programme.