



# A modular library for fast prototyping of solution-state nuclear magnetic resonance experiments

Michał Górka and Wiktor Koźmiński

Biological and Chemical Research Centre, Faculty of Chemistry,  
University of Warsaw, Żwirki i Wigury 101, 02-089 Warsaw, Poland

**Correspondence:** Wiktor Koźmiński (kozmin@chem.uw.edu.pl)

Received: 29 January 2024 – Discussion started: 31 January 2024

Revised: 20 March 2024 – Accepted: 20 March 2024 – Published: 23 April 2024

**Abstract.** We present a framework library (Modular Elements, ME) for the development of pulse sequences for Bruker spectrometers. It implements a two-level abstraction approach – the lower level comprises basic functional elements of pulse sequences and the higher one often-reused blocks comprising several evolution periods. The low-level abstractions reduce code duplication between variants of experiments such as hard-pulse and selective variants of individual NMR experiments. The high-level modules enable further reuse of pulse program code and aid in the construction of complex experiments. We show the library's functionality by presenting pulse programs that can be switched between standard and TROSY variants as well as hard and shaped pulses and that can seamlessly incorporate real-time homodecoupling. Adaptability is further demonstrated in a configurable 4D NOESY program.

## 1 Introduction

NMR is an extraordinarily powerful and adaptable spectroscopic method, with just the solution-state variant being capable of discerning the structure and dynamics of molecules ranging in size from simple organic compounds to large protein complexes such as a proteasome (Sprangers and Kay, 2007). The variety of experimental objects and the great number of parameters that can be measured have led to the proliferation of not only general experimental schemes, such as an  $^1\text{H}$ ,  $^{15}\text{N}$  HSQC (Bodenhausen and Ruben, 1980) or a HNC0 (Kay et al., 1990b; Ikura et al., 1990), but also their variants and thus the pulse sequences, implementing them as computer code. As an example, for the often-used HNC0 experiment, the non-exhaustive list of meaningful implementation choices is as follows: the experiment can use hard pulses or avoid saturating water using selective pulses (Schanda et al., 2006); the final transfer element can be a simple spin echo (Palmer et al., 1991), a set of three echoes implementing a sensitivity-enhanced transfer or one of many TROSY variants (Salzmann et al., 1999b; Nietlispach, 2005), with possible optimisations (Salzmann et al., 1999a; Schulte-Herbrüggen and Sørensen, 2000); and radiation damping

can be suppressed with bipolar gradients (Sklenar, 1995). Even without implementing all specialised experiment variants, the standard library supplied with the TopSpin software (Bruker) contains over 1000 pulse programs.

A common problem with pulse sequences, especially in biological NMR, is thus the requirement to code multiple variants of a given sequence. If this is done in separate files (as in the TopSpin built-in library), it results in a lot of code repetition, and, if done using conditional statements, it can substantially complicate the structure of the file, making troubleshooting harder. Similarly, many pulse sequences share large amounts of code, often with no or minimal changes. Because this repeated code is scattered across different sequences and variants of experiments, adding new variants (using different soft pulses or adding homodecoupling) requires the same modification to be applied across a large part of the whole pulse sequence library, which is tedious and error-prone. It is possible to implement such a library using a standard systems programming language like C or Python, but we decided to use the native programming language of the spectrometer system, since any user writing a pulse sequence needs to be familiar with it, and requiring knowledge

of a separate programming language and its tooling would be an unnecessary hurdle to adoption. Here we show that, by abstracting certain functionalities using the somewhat limited macro and define statements built into the TopSpin software, the problems described above can still be avoided and the code can be made more readable and easier to modify. Here, we present the Modular Elements (ME) library for Bruker spectrometers. Although the library is specific to a particular hardware vendor, the modular approach it implements is more general and can be implemented on other instruments. A previous implementation of a modular library for pulse program implementation (NMR blocks) can be found in Zawadzka-Kazimierczuk (2012) for Varian/Agilent spectrometers, where evolution periods such as INEPT or COS-INEPT were abstracted as C functions. Alternative approaches to a modular library include domain-specific pulse program generators, like GENESIS (Yong et al., 2022) for NOAH supersequences. Specialised libraries combining custom pulse programs and various tools (Favier and Brutscher, 2019; Vallet et al., 2020; Lukavsky and Puglisi, 2001) are suitable for routine use but have limited applicability in the prototyping of new sequences.

## 2 General approach to pulse sequence modularisation

We categorise the library's functionality as low and high level. Low-level functionality encompasses the creation of variables and functions (technically functional macros), abstracting the basic building blocks of pulse sequences like pulses, gradients and delays. A pulse function can expand to a 90° proton hard pulse or an HN selective excitation pulse, depending on the global settings. A gradient function can expand to “no operation” in standard HSQC or a selection gradient in the gradient-selected HSQC variant, with its corresponding delay variable containing a zero or correspondingly non-zero length of time. Decoupling functions for protons and deuterium can likewise be enabled and disabled depending on whether a TROSY variant is desired and whether the sample is deuterated. This functionality simplifies the writing of pulse sequences implementing multiple variants of a given NMR experiment and gives its user the ability to easily test and compare the effectiveness of the variants for a given sample, and the commonisation of parameters across variants enables faster optimisation.

High-level functionality is implemented as modules that are included in the pulse sequences and that can be classified as general modules and specific modules. General modules implement elements common to almost all pulse sequences. The functionally most significant ones are the preparation and acquisition modules. The preparation module gives the user the option of turning on functionality such as solvent presaturation or a combination of N/C pulses and pulse field gradients for spoiling the residual magnetisation on those nu-

clei. The acquisition module enables switching between standard or homodecoupled data acquisition. The specific modules are abstract blocks of pulse sequence elements that appear in many pulse sequences in an almost identical form. The two main types of specific modules are proximal and distal modules, abstracting the functionality of blocks including and following first excitation (distal) and right before acquisition (proximal). Despite a large variety of possible implementations, the proximal and distal fragments differentiate between variants of a pulse sequence (e.g. a standard hard-pulse HNCO, selective or BEST HNCO, or a hard-pulse and BEST TROSY-HNCO; Solyom et al., 2013), and the actual code is usually repeatable across different sequences. HNCO, HNCACO and HNCOCA (Yang and Kay, 1999) have very similar proximal and distal parts; HN(CA)CONH and HabCabCONH (Kazimierczuk et al., 2010) have different proximal blocks, but the distal block is still very similar for all the sequences listed. With the use of the low-level functionality described above, a single proximal module can abstract the initial two transfer periods (first with transverse H magnetisation and second with transverse N/C magnetisation), with the choice of N/C nucleus and the choice of evolved  $J$  coupling (CO in HNCO) made using a define directive in the main pulse sequence. NOESY experiments are particularly susceptible to modularisation, with the NOE transfer period naturally splitting them into proximal and distal blocks. Standard 2D experiments of the HSQC, TROSY and HMQC types have thus been implemented as proximal modules that can be used on their own as 2D experiments or included in a 3D or 4D NOESY (Kay et al., 1990a) with the chosen distal modules, which can themselves be modified 2D experiments or simpler blocks.

## 3 Library implementation

Description of implementation details and design choices requires a quick recapitulation of TopSpin pulse programs' language specifics. TopSpin has two types of variables: user-adjustable numbered variables ( $d1-d63$  for delays,  $cnst1-cnst63$  for floating point constants and similar for integer constants (“loopcounters”)  $lN$ , pulse lengths  $pN$ , etc.) and named variables (pulses, delays and loopcounters only, as well as lists of various kinds), which can only be manipulated within a pulse program. Some less-documented observations on the limitations of named variables are compiled in the library documentation (Górka and Koźmiński, 2024a). TopSpin implements limited functionality for defining text-substitution macros (the “-traditional” mode of the GNU C preprocessor `cpp`; Stallman and GCC Developer Community, 2012), which can be used everywhere outside a “relation” (variable value calculations using a subset of C syntax) due to their implementation as text in quotes (treated as string literals by `cpp` and ignored for macro expansion), though this limitation can be overcome (see the file “notes on Top-

Spin.txt” in the ME library). The user can provide custom option choices to a pulse program using the ZGOPNTS variable to define appropriate macros.

### 3.1 Low-level modularisation

#### 3.1.1 Variables

With no user-adjustable named variables, two approaches to making them consistent across different pulse programs are possible – indirection through named variables or introduction of a convention attaching constant meanings to numbered variables. Due to the limited number and type of named variables, we predominantly use the latter option (with sets of variables described in files such as delays.incl, pulse.incl, etc.) with some focused use of indirection – for example, proximal-type modules use *timeHX* and *timeXY* for *J* coupling evolution times between the H, X and Y nuclei. Default values for all such variables can be set using the `me.set_parameters.py` TopSpin program. For variables that do not ordinarily have calculations performed on them (pulse phases *phN*, gradient programs *gpN*), we implemented full indirection, where the user can use `phFree1` or `phFree3` without worrying about which *phN* variables are used by other parts of a pulse program.

#### 3.1.2 Pulses

The most important low-level abstractions are pulse functions. They are implemented using function-like macros of cpp and have the general form `nucleus_type(phase)`, where “nucleus” can be a general specifier like H/C/N or more specific like HN/HC/CA/CO and type is classified based on the desired functionality, with the main ones being excitation (for the excitation of longitudinal magnetisation), flipback (acting on transverse magnetisation), refocusing and inversion (inverting longitudinal magnetisation) and `excitation_UR` and `flipback_UR` (implementing universal rotations). The pulse macros will have a different replacement text based on global settings (usually ZGOPTNS). A proton pulse “`H_excitation(ph)`” will be replaced with a hard pulse “`p1 ph p11`” by default, but a “`-DH_SHAPED`” option will instead be replaced with “`p54:sp54 ph`” for a selective soft pulse and the associated named variable “`pH_excitation`” will be set to have the same value as *p1* or *p54*.

Pulse programs should account for the effective evolution time during pulses (which can be as much as 1 ms for longer selective pulses) to give correctly phased spectra and optimal *J*-coupling evolution times. This library only accounts for linear phase slopes using the modelling method described in Lescop et al. (2010), i.e. treating a pulse as a sequence (delay, ideal pulse, delay), which accounts for the phase slope of many commonly used pulses and which can be explicitly optimised during pulse design (Gershenson et al., 2008; Asami et al., 2018). This phase slope is compensated for using variables such as *eH\_excitation*, which for the hard pulse above

would be set to  $\frac{2p1}{\pi}$ . We assume that the flipback and flipback\_UR pulses act as if they were time-reversed excitation pulses, and so the effective evolution time for a flipback pulse acting on transverse magnetisation is also *eH\_excitation*. A `H_excitation_UR` pulse of phase *x* will give effective times of *eH\_excitation* for *z* magnetisation, *eH\_flipback* for *y* magnetisation and *eH\_excitation* + *eH\_flipback* for *x* magnetisation. By compensating for delays using the above-mentioned variables, the whole sequence can be switched from a hard-pulse implementation to a shaped-pulse version to account for field inhomogeneity or to perform band-selective excitation.

#### 3.1.3 Code blocks

There are many small blocks of code that can be included or excluded in a pulse program based on a sequence variant. To limit the number of conditional statements in the main pulse program, many are defined as macros that will expand to pulse program code based on options. For example, “`H2O_FLIPBACK(ph2)`” will be replaced with “`(11:sp1 ph2):f1`” in a pulse sequence with water flipback and with whitespace when using selective pulses. Similarly, `DECOUPLE_H_ON` and `DECOUPLE_H_OFF` macros will turn on proton decoupling in a standard HNC0 experiment but will have no effect in TROSY-HNC0.

### 3.2 High-level modularisation

TopSpin pulse programs follow a defined sequential structure that complicates the implementation of high-level modules as individual files and that, in general, consists of

1. configuration and compile-time calculations,
2. a “`zd`” or “`ze`” statement,
3. a pulse program body (pulses and delays) and real-time calculations,
4. a signal acquisition block,
5. loop statements for scans of a free induction decay (FID) and points of a multi-dimensional experiment, and
6. phase program definitions.

#### 3.2.1 General modules

The general modules fit into this sequential structure as follows.

- 1a. Configuration and compile-time calculations
- 1b. `init.incl`
- 1c. Configuration and compile-time calculations continued

2. A “zd” or “ze” statement
- 3a. Real-time calculations
- 3b. start.incl
- 3c. A pulse program body (pulses and delays) and real-time calculations
4. end.incl
5. Loop statements for scans of a FID and points of a multi-dimensional experiment
- 6a. phasecycles.incl
- 6b. Phase program definitions

The general modules have numerous conditional statements and imports evaluating the option provided in point (1) above and using the built-in ZGOPTNS variable to interact with the specific modules (this is covered below). The init.incl module provides the library’s core functionality by defining macros for functions and variable descriptions; start.incl executes the relaxation delay (with possible solvent presaturation) and optional operations, such as crushing residual C or N magnetisation (gradient pulse after an excitation pulse) or inverting N magnetisation before the relaxation delay in BEST-TROSY. For non-protein experiments, an acceleration by sharing adjacent polarization (ASAP) (Kupče and Freeman, 2007) period would be added here, but the relevant code is experimental and provided in a commented-out form due to the method’s potential to damage probe heads. The end.incl module handles acquisition with the option for real-time homodecoupling – here provided with the  $^{13}\text{C}$ -GBIRD<sup>r,X</sup> (Garbow et al., 1982; Haller et al., 2022) and BASHD (Brüschweiler et al., 1988; Krishnamurthy, 1997) types.

### 3.2.2 Specific modules

In contrast to the general modules, specific modules implement a specific form of a proximal or distal block and serve to localise the relevant code in a single file. The biggest hurdle to writing self-contained modules for TopSpin is the sequential pulse program structure necessitating the separation of related code segments in the post-preprocessing file. To mitigate this problem, each module is entirely enclosed in a conditional statement with alternative conditions (an if...elif...else structure), and including the file once will only insert a selected part of the module into a file. Since the four general modules already perform the sequential separation of code, each of them sets the appropriate conditions (defines a macro) and imports distal\_2D.incl and proximal\_2D.incl, which themselves import the selected specific modules at each of the four positions in the pulse program. Thus, the initialisation-phase statements (variable declarations, some calculations, macro definitions) are included in

init.incl, and runtime calculations of both types of modules are included through start.incl together with the main body (pulses and delay statements) of the distal block. Similarly, the main body of the proximal module is included through end.incl before the latter’s acquisition portion. Phase cycles of both modules are inserted into a pulse program file through phasecycles.incl with some basic logic, allowing for coordination of the cycles between them if two modules are used.

For triple-resonance experiments (in the implementation limited to amide protons, but it should be possible to extend them to aliphatic and aromatic groups), the proximal module hx.incl and the distal module hx.incl provide the ability to compartmentalise the relatively standard blocks for both out-and-back- and straight-through-type experiments, and a more detailed description in the context of a HNCQ experiment is provided below. Although sub-optimal under some circumstances, the library provides default two-step phase cycles for each of the modules, leaving the implementation of eight-step and longer cycles for the central part of the program. A more detailed description of the individual modules is provided in the library documentation. In the Supplement we provide a detailed step-by-step description of the proximal HSQC module and the way it is used in the 2D experiment pulse program.

A specific module separate from the proximal and distal types can also be based on the same structure and manually included either in the pulse program after each general module or in a specific module itself – the se.incl module implements the sensitivity-enhanced COS-INEPT and TROSY transfers and is imported into both the hsqc\_se.incl and hx.incl modules.

## 4 Application examples

### 4.1 HNCQ

HNCQ is one of the simplest triple-resonance experiments and thus a good candidate for demonstrating the strengths and limitations of the presented approach to library building. We present its ME NMR implementation in Fig. 1. We use a custom prosol file (used mostly for automatic precalculation of pulse parameters) to free up a number of variables. Evolution delays and increments are defined explicitly due to the proximal module’s numbered variables (here *td2* and *in2*) being dimensionality-dependent. The block of definitions specifies options for the ME library – the proximal (xh.incl) and distal (hx.incl) modules and the couplings to be evolved (*Y* is  $^2J_{\text{NCO}}$ ) and decoupled (*A* is  $^2J_{\text{NCA}}$ ). After importing the first two general modules, which include the distal modules and two evolution periods, the carbonyl echo is implemented using the library’s low-level functionality. Since channels and pulses are not selected explicitly, this block will function with split CA and CO channels (with the right spectrometer configurations and “CACO\_SPLIT” defined in ZGOPTNS) or use a single carbon channel and frequency-offset pulses.

```

prosol relations=<me>

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

# define DIMS 3

/*Select options for distal and proximal blocks:*/
# define XH
# define HX
# define DISTAL_N
# define DISTAL_Y_CO
# define DISTAL_A_CA
# define PROXIMAL_NH
# define PROXIMAL_Y_CO
# define PROXIMAL_A_CA

; Variable definitions for the distal (H->N) and proximal (N->H) blocks:
# include <ME/includes/init.incl>

1 ze

; Relaxation and distal block Hz -> NzHz -> COzNz:
# include <ME/includes/start.incl>

; 2COzNz CO evolution (T1):
(CO_excitation(phFree1)):fco
T1*0.5
(center (CA_CO_inversion(ph0)):fCA (N_inversion(ph0)):fn)
T1*0.5
(CO_refocussing(ph0)):fco
(CA_CO_inversion(ph0)):fCA ; BSP compensation.
(CO_flipback(ph0)):fco

GRAD(gpFree1)

; Proximal block COzNz -> NzHz-> H and acquisition:
# include <ME/includes/end.incl>
FIPH(calph(phFree1,+90), caldel(T1, +in1))
exit

# include <ME/includes/phasecycles.incl>

phFree1 = 0 0 0 0 2 2 2 2

; Receiver phase:
phRec = PROXIMAL_PH31 + DISTAL_PH31 + phFree1

;gpzFree1: gradient after CO echo: 21%.
;gpnamFree1: SMSQ10.100

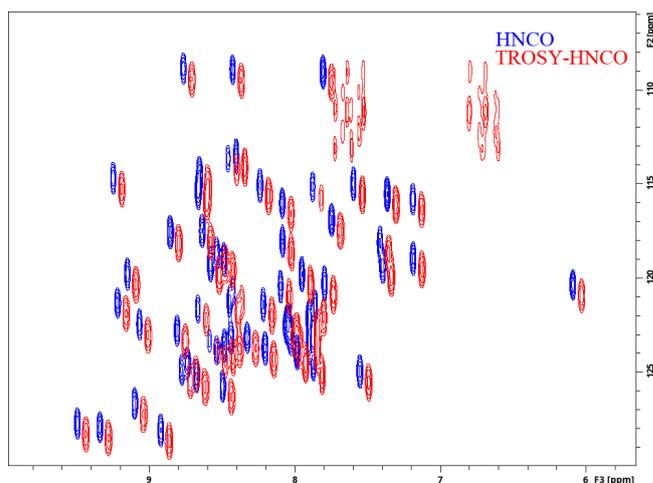
```

**Figure 1.** Pulse program code for the implementation of the HNCO experiment.

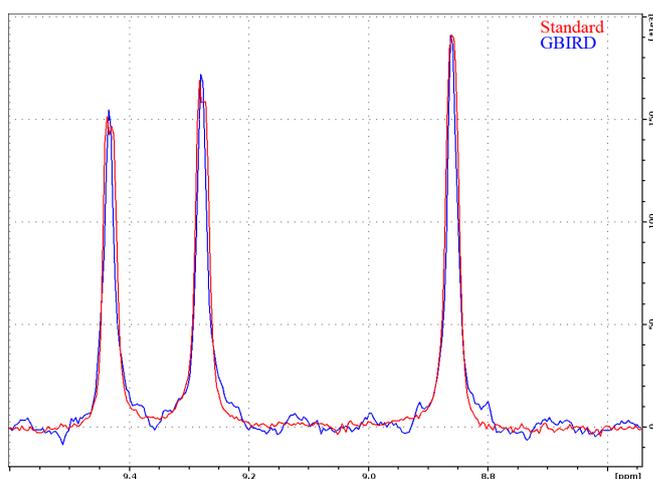
The rest of the pulse program includes the end.incl module (with the two proximal echoes and the acquisition) and standard configuration of gradients and phase cycles. To demonstrate the library functionality, in Fig. 2 we present 2D spectra (recorded as HN(CO) experiments) of a standard variant of the experiment (no ZGPTNs) and a TROSY-HNCO (adding the TROSY definition to ZGPTNs), selecting only the  $H_\beta$  and  $N_\beta$  components (the lower-right component uses a standard display convention). It is possible to choose a  $^{13}\text{C}$ -GBIRD $^{r,X}$  by appending the “ACQ\_BIRD\_C” option to ZGPTNS, with an example of line narrowing demonstrated in Fig. 3.

#### 4.2 Four-dimensional NOESY

The modular nature of the library is exemplified by the 4D NOESY pulse program in Fig. 4. Apart from the basic structure described above in the case of HNCO, it only contains a mixing period joining the proximal and distal modules, with the evolved heteronuclei and experiment types selected by



**Figure 2.** Experimental demonstration of the implementations of the HNCO and TROSY-HNCO experiments for ubiquitin 8 (kDa) at 25 °C. Spectra were recorded as  $^1\text{H}$ - $^{15}\text{N}$  planes with maximum evolution times of 85.2 ms ( $^1\text{H}$ ) and 9.87 ms ( $^{15}\text{N}$ ) and processed using cosine-squared window functions.



**Figure 3.** One-dimensional slices (for  $N = 128.5$  ppm) through  $^1\text{H}$ - $^{15}\text{N}$  planes recorded for TROSY-HNCO with standard acquisition and TROSY-HNCO with  $^{13}\text{C}$ -GBIRD $^{r,X}$ , demonstrating the effectiveness of the homodecoupling and the resultant line narrowing. Both spectra were acquired for ubiquitin 8 (kDa) at 25 °C with maximum evolution times of 340.7 ms ( $^1\text{H}$ ) and 9.87 ms ( $^{15}\text{N}$ ), processed using a cosine-squared window function in the N dimension and sine-squared-shifted by  $\frac{\pi}{2}$  in the H dimension. The GBIRD spectrum was shifted right by 4 Hz (a shift was possibly induced by sample heating) and scaled up to match the amplitude of the standard TROSY-HNCO. For the GBIRD spectrum, 18 chunks were acquired with a 11.96 ms inter-chunk delay, a 3.5 ms  $^2J_{\text{HC}}$  evolution time and a 120  $\mu\text{s}$  BIP-720-100-10 (Smith et al., 2001) pulse for  $^{13}\text{C}$  inversion. Line widths at half height are (from left to right) 19.6, 19.5 and 19.1 Hz for the standard spectrum and 13.2, 13.2 and 13.7 Hz for the homodecoupled spectrum (TopSpin peak function).

```

proscl relations=<me>

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

# define NOESY
# define DIMS 4

; Variable definitions and calculations for the proximal and distal 2Ds:
# include <ME/includes/init.incl>

define delay mixTime
;d10: NOESY mixing time [40-400 ms]
"mixTime = d10 - pGRAD - dGRAD" ; Corrected for gradient.

1 ze

; Distal 2D:
# include <ME/includes/start.incl>

; NOESY mixing:
# ifdef MIX_LOCKED
(
  refAlign (mixTime):fH
  lAlign (1m 4u BLKGRAD):fH
  rAlign (2m UNBLKGRAD):fH
)
# else
  mixTime
# endif

GRAD(gpNOESY)

; Proximal 2D and acquisition:
# include <ME/includes/end.incl>
exit

# include <ME/includes/phasecycles.incl>

; Receiver phase:
phRec = PROXIMAL_PH31 + DISTAL_PH31
;gpzNOESY: gradient after NOESY: -7%.
;gpnamNOESY: SMSQ10.100

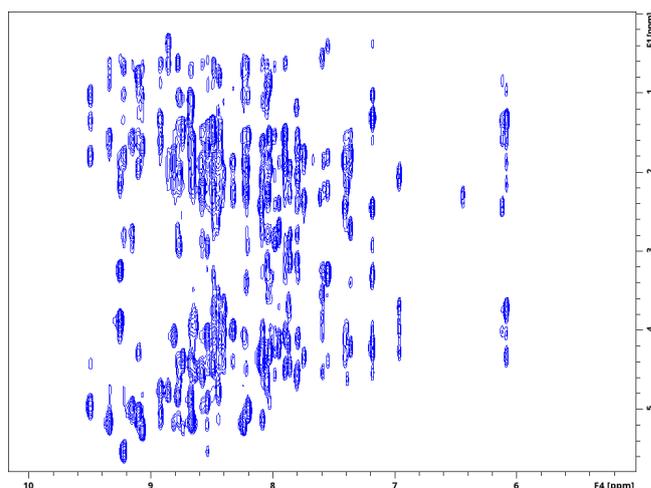
```

**Figure 4.** Pulse program code for the implementation of a 4D NOESY experiment.

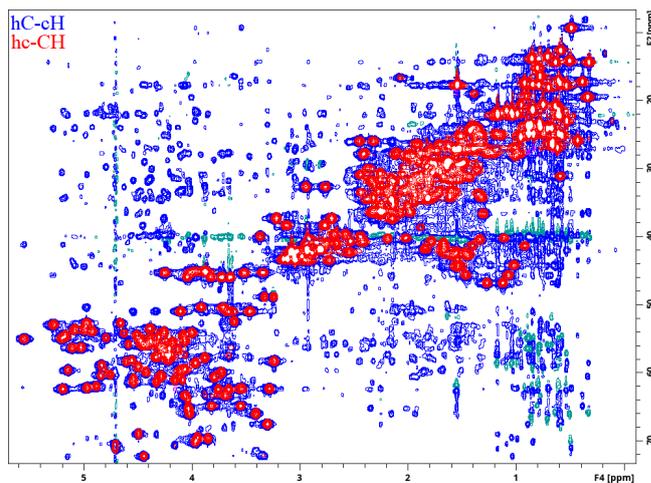
the user with ZGOPTNS. HC,NH-HMQC-NOESY-HSQC with sensitivity enhancement in the last dimension (Fig. 5) can be changed to a HC,CH-HMQC-NOESY-HSQC (Fig. 6) pulse program by changing the “PROXIMAL\_N” option to “PROXIMAL\_C” and adding the gradient selection option (“GS”), which is not a default for non-sensitivity-enhanced HSQC.

### 4.3 $^1\text{H}$ - $^{15}\text{N}$ correlation – shaped pulses

Since BEST-type experiments utilizing shaped pulses can bring improved sensitivity, especially at higher scan repetition rates (Schanda et al., 2006), we demonstrate the library’s inherent ability to automatically adapt to the substantial chemical shift and coupling evolution during the  $90^\circ$  universal-rotation E400B (Veshtort and Griffin, 2004) (using a time-reversed version of the original pulse for excitation) pulses with a length of  $1073.1\ \mu\text{s}$  (equivalent to an ideal pulse followed by a  $611.7\ \mu\text{s}$  delay) and refocusing pulse REBURP (Geen and Freeman, 1991) with a length of  $1108.8\ \mu\text{s}$  (modelled as an ideal refocusing pulse flanked by  $554\ \mu\text{s}$  delays) in Figs. 7 and 8. With a relaxation delay of  $0.65\ \text{s}$ , all peaks in the selected region are over 20% stronger in the shaped-pulse version. Full datasets for a number of differ-



**Figure 5.**  $^1\text{H}$ - $^1\text{H}$  planes recorded using a 4D HC,NH-HMQC-NOESY-HSQC experiment for ubiquitin 8 (kDa) at  $25^\circ\text{C}$ . Spectra were recorded with maximum evolution times of  $85.2\ \text{ms}$  ( $^1\text{H}$  direct dimension) and  $6.99\ \text{ms}$  ( $^1\text{H}$  indirect dimension) and processed using cosine-squared window functions.

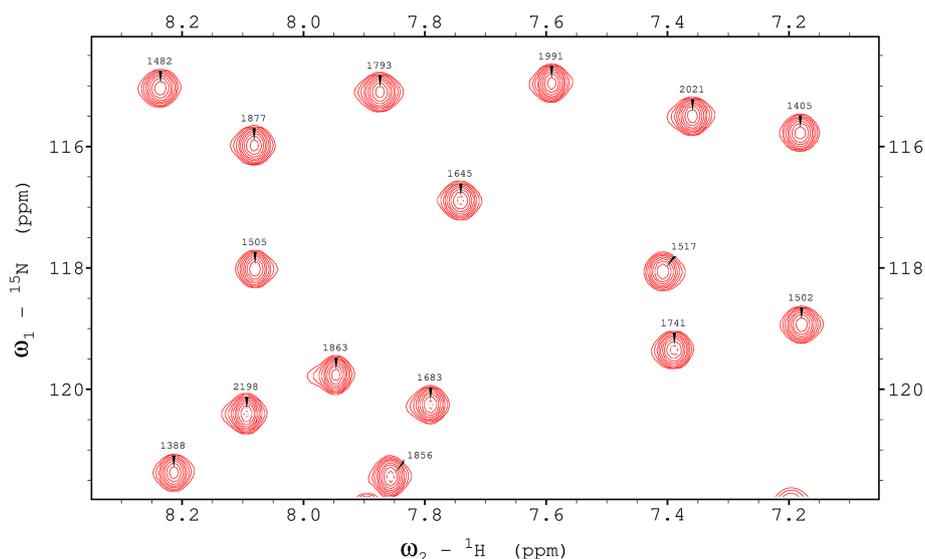


**Figure 6.** Two different  $^1\text{H}$ - $^{13}\text{C}$  2D planes recorded using a ME implementation of a 4D HC,CH-HMQC-NOESY-HSQC experiment for ubiquitin 8 (kDa) at  $25^\circ\text{C}$ . Spectra were recorded with maximum evolution times of  $85.2\ \text{ms}$  ( $^1\text{H}$  direct dimension) and  $7.96\ \text{ms}$  (both  $^{13}\text{C}$  dimensions) and processed using cosine-squared window functions.

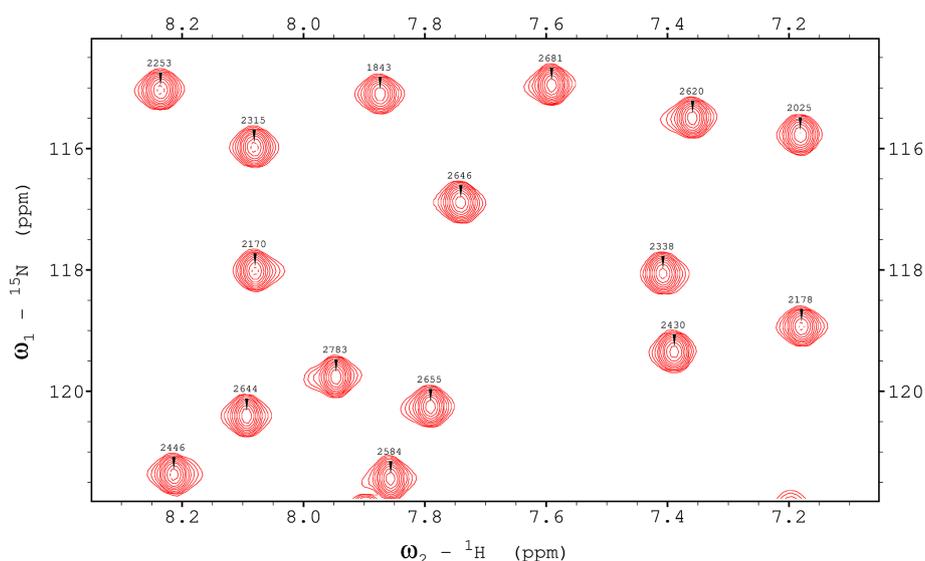
ent relaxation delays are provided, as in the Data availability section.

## 5 Materials and methods

For all the experiments, we used a  $2\ \text{mM}$   $^{13}\text{C}$ , $^{15}\text{N}$  double-labelled human ubiquitin (ASLA Biotech) in a  $5\ \text{mm}$  Shigemi NMR microtube. All the spectra were acquired using a Bruker Avance IIIHD  $800\ \text{MHz}$  spectrometer with a  $5\ \text{mm}$  TCI  $z$ -gradient cryoprobe. Pulse lengths for  $90^\circ$  hard



**Figure 7.**  $^1\text{H}$ ,  $^{15}\text{N}$  TROSY spectrum recorded using a ME implementation with hard pulses and water flipback for ubiquitin 8 (kDa) at 25 °C. The spectrum was recorded with maximum evolution times of 85.2 ms ( $^1\text{H}$ ) and 39.5 ms ( $^{15}\text{N}$ ) and processed using cosine-squared window functions.



**Figure 8.**  $^1\text{H}$ ,  $^{15}\text{N}$  TROSY spectrum recorded using a ME implementation with shaped pulses (E400B and RE-BURP) for ubiquitin 8 (kDa) at 25 °C. The spectrum was recorded with maximum evolution times of 85.2 ms ( $^1\text{H}$ ) and 39.5 ms ( $^{15}\text{N}$ ) and processed using cosine-squared window functions.

pulses were 10.47  $\mu\text{s}$  for  $^1\text{H}$ , 12.3  $\mu\text{s}$  for  $^{13}\text{C}$  and 33.22  $\mu\text{s}$  for  $^{15}\text{N}$ . Full acquisition and processing parameters are provided in the dataset linked below in the Data availability section. Acquisition and library testing were performed using the TopSpin 3.6.5 Service Pack 2 software (Bruker). Data processing and plotting (aside from Figs. 7 and 8) were carried out in TopSpin. Figures 7 and 8 were prepared using the NMRFAM-SPARKY software (Goddard and Kneller, 2004; Lee et al., 2015).

## 6 Conclusions

We have described a framework library implementing a two-level approach to pulse program modularisation and demonstrated its utility. We hope it can be used by others either directly for the streamlining of pulse program code or as an inspiration for similar frameworks. Although the usefulness of the modularisation approach is most obvious for the case of the protein experiments presented here, it should extend to nucleic acids and, to a more limited extent, small molecules.

In the latter case, the ability to modularise preparation period operations (presaturation, ASAP), WATERGATE-type (Piotto et al., 1992; Sklenar et al., 1993) solvent suppression and real-time acquisition should be particularly useful.

**Code availability.** The initial version of the ME library code and documentation is available online at <https://doi.org/10.5281/zenodo.10841749> (Górka and Koźmiński, 2024a). The current library version is available online at [https://github.com/nmr-cnbc/MENmr\\_pubcode](https://github.com/nmr-cnbc/MENmr_pubcode) (last access: 20 March 2024) or from the authors upon request.

**Data availability.** All data used in the preparation of this article are available online at <https://doi.org/10.5281/zenodo.10578330> (Górka and Koźmiński, 2024b).

**Supplement.** The supplement related to this article is available online at: <https://doi.org/10.5194/mr-5-51-2024-supplement>.

**Author contributions.** MG and WK designed the general workflow of the ME library. MG wrote the library code and performed the experiments. MG wrote the manuscript with input from WK.

**Competing interests.** The contact author has declared that neither of the authors has any competing interests.

**Disclaimer.** Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

**Financial support.** This research was supported by Polish National Science Centre grant PRELUDIUM 2015/19/N/ST4/00863 to Michał Górka.

**Review statement.** This paper was edited by Patrick Giraudeau and reviewed by Eriks Kupce and Bernhard Brutscher.

## References

Asami, S., Kallies, W., Günther, J. C., Stavropoulou, M., Glaser, S. J., and Sattler, M.: Ultrashort Broadband Cooperative Pulses for Multidimensional Biomolecular NMR Experiments, *Angew. Chem.*, 130, 14706–14710, <https://doi.org/10.1002/ange.201800220>, 2018.

Bodenhausen, G. and Ruben, D. J.: Natural abundance nitrogen-15 NMR by enhanced heteronuclear spectroscopy, *Chem. Phys. Let-*

*ters*, 69, 185–189, [https://doi.org/10.1016/0009-2614\(80\)80041-8](https://doi.org/10.1016/0009-2614(80)80041-8), 1980.

- Brüschweiler, R., Griesinger, C., Sørensen, O. W., and Ernst, R. R.: Combined use of hard and soft pulses for  $\omega_1$  decoupling in two-dimensional NMR spectroscopy, *J. Magn. Reson.*, 78, 178–185, [https://doi.org/10.1016/0022-2364\(88\)90171-0](https://doi.org/10.1016/0022-2364(88)90171-0), 1988.
- Favier, A. and Brutscher, B.: NMRlib: user-friendly pulse sequence tools for Bruker NMR spectrometers, *J. Biomol. NMR*, 73, 199–211, <https://doi.org/10.1007/s10858-019-00249-1>, 2019.
- Garbow, J. R., Weitekamp, D. P., and Pines, A.: Bilinear rotation decoupling of homonuclear scalar interactions, *Chemical Phys. Lett.*, 93, 504–509, [https://doi.org/10.1016/0009-2614\(82\)83229-6](https://doi.org/10.1016/0009-2614(82)83229-6), 1982.
- Geen, H. and Freeman, R.: Band-selective radiofrequency pulses, *J. Magn. Reson.*, 93, 93–141, [https://doi.org/10.1016/0022-2364\(91\)90034-Q](https://doi.org/10.1016/0022-2364(91)90034-Q), 1991.
- Gershenson, N. I., Skinner, T. E., Brutscher, B., Khaneja, N., Nimbalkar, M., Luy, B., and Glaser, S. J.: Linear phase slope in pulse design: Application to coherence transfer, *J. Magn. Reson.*, 192, 235–243, <https://doi.org/10.1016/j.jmr.2008.02.021>, 2008.
- Goddard, T. D. and Kneller, D. G.: SPARKY 3, University of California, San Francisco, 2004.
- Górka, M. and Koźmiński, W.: ME library, Zenodo [code], <https://doi.org/10.5281/zenodo.10841749>, 2024a.
- Górka, M. and Koźmiński, W.: Sample data for ME library, Zenodo [data set], <https://doi.org/10.5281/zenodo.10578330>, 2024b.
- Haller, J. D., Bodor, A., and Luy, B.: Pure shift amide detection in conventional and TROSY-type experiments of  $^{13}\text{C}$ ,  $^{15}\text{N}$ -labeled proteins, *J. Biomol. NMR*, 76, 213–221, <https://doi.org/10.1007/s10858-022-00406-z>, 2022.
- Ikura, M., Kay, L. E., and Bax, A.: A novel approach for sequential assignment of  $^1\text{H}$ ,  $^{13}\text{C}$ , and  $^{15}\text{N}$  spectra of proteins: heteronuclear triple-resonance three-dimensional NMR spectroscopy. Application to calmodulin, *Biochemistry*, 29, 4659–4667, 1990.
- Kay, L. E., Clore, G. M., Bax, A., and Gronenborn, A. M.: Four-dimensional heteronuclear triple-resonance NMR spectroscopy of interleukin-1 beta in solution, *Science*, 249, 411–414, <https://doi.org/10.1126/science.2377896>, 1990a.
- Kay, L. E., Ikura, M., Tschudin, R., and Bax, A.: Three-dimensional triple-resonance NMR spectroscopy of isotopically enriched proteins, *J. Magn. Reson.*, 89, 496–514, [https://doi.org/10.1016/0022-2364\(90\)90333-5](https://doi.org/10.1016/0022-2364(90)90333-5), 1990b.
- Kazimierczuk, K., Zawadzka-Kazimierczuk, A., and Koźmiński, W.: Non-uniform frequency domain for optimal exploitation of non-uniform sampling, *J. Magn. Reson.*, 205, 286–292, <https://doi.org/10.1016/j.jmr.2010.05.012>, 2010.
- Krishnamurthy, V. V.: Application of Semi-Selective Excitation Sculpting for Homonuclear Decoupling During Evolution in Multi-Dimensional NMR, *Magn. Reson. Chem.*, 35, 9–12, [https://doi.org/10.1002/\(SICI\)1097-458X\(199701\)35:1<9::AID-OMR930>3.0.CO;2-R](https://doi.org/10.1002/(SICI)1097-458X(199701)35:1<9::AID-OMR930>3.0.CO;2-R), 1997.
- Kupče, E. and Freeman, R.: Fast multidimensional NMR by polarization sharing, *Magn. Reson. Chem.*, 45, 2–4, <https://doi.org/10.1002/mrc.1931>, 2007.
- Lee, W., Tonelli, M., and Markley, J. L.: NMRFAM-SPARKY: enhanced software for biomolecular NMR spectroscopy, *Bioinformatics*, 31, 1325–1327, <https://doi.org/10.1093/bioinformatics/btu830>, 2015.

- Lescop, E., Kern, T., and Brutscher, B.: Guidelines for the use of band-selective radiofrequency pulses in hetero-nuclear NMR: Example of longitudinal-relaxation-enhanced BEST-type  $^1\text{H}$ – $^{15}\text{N}$  correlation experiments, *J. Magn. Reson.*, 203, 190–198, <https://doi.org/10.1016/j.jmr.2009.12.001>, 2010.
- Lukavsky, P. J. and Puglisi, J. D.: RNAPack: An Integrated NMR Approach to RNA Structure Determination, *Methods*, 25, 316–332, <https://doi.org/10.1006/meth.2001.1244>, 2001.
- Nietlispach, D.: Suppression of anti-TROSY lines in a sensitivity enhanced gradient selection TROSY scheme, *J. Biomol. NMR*, 31, 161–166, <https://doi.org/10.1007/s10858-004-8195-7>, 2005.
- Palmer, A. G., Cavanagh, J., Wright, P. E., and Rance, M.: Sensitivity improvement in proton-detected two-dimensional heteronuclear correlation NMR spectroscopy, *J. Magn. Reson.*, 93, 151–170, [https://doi.org/10.1016/0022-2364\(91\)90036-S](https://doi.org/10.1016/0022-2364(91)90036-S), 1991.
- Piotto, M., Saudek, V., and Sklenář, V.: Gradient-tailored excitation for single-quantum NMR spectroscopy of aqueous solutions, *J. Biomol. NMR*, 2, 661–665, <https://doi.org/10.1007/BF02192855>, 1992.
- Salzmann, M., Wider, G., Pervushin, K., and Wüthrich, K.: Improved sensitivity and coherence selection for  $[\text{15N}, \text{1H}]$ -TROSY elements in triple resonance experiments, *J. Biomol. NMR*, 15, 181–184, <https://doi.org/10.1023/A:1008358030477>, 1999a.
- Salzmann, M., Wider, G., Pervushin, K., Senn, H., and Wüthrich, K.: TROSY-type Triple-Resonance Experiments for Sequential NMR Assignments of Large Proteins, *J. Am. Chem. Soc.*, 121, 844–848, <https://doi.org/10.1021/ja9834226>, 1999b.
- Schanda, P., Van Melckebeke, H., and Brutscher, B.: Speeding Up Three-Dimensional Protein NMR Experiments to a Few Minutes, *J. Am. Chem. Soc.*, 128, 9042–9043, <https://doi.org/10.1021/ja062025p>, 2006.
- Schulte-Herbrüggen, T. and Sørensen, O. W.: Clean TROSY: Compensation for Relaxation-Induced Artifacts, *J. Magn. Reson.*, 144, 123–128, <https://doi.org/10.1006/jmre.2000.2020>, 2000.
- Sklenar, V.: Suppression of Radiation Damping in Multi-dimensional NMR Experiments Using Magnetic Field Gradients, *J. Magn. Reson., Series A*, 114, 132–135, <https://doi.org/10.1006/jmra.1995.1119>, 1995.
- Sklenar, V., Piotto, M., Leppik, R., and Saudek, V.: Gradient-Tailored Water Suppression for  $^1\text{H}$ - $^{15}\text{N}$  HSQC Experiments Optimized to Retain Full Sensitivity, *J. Magn. Reson., Series A*, 102, 241–245, <https://doi.org/10.1006/jmra.1993.1098>, 1993.
- Smith, M. A., Hu, H., and Shaka, A. J.: Improved Broadband Inversion Performance for NMR in Liquids, *J. Magn. Reson.*, 151, 269–283, <https://doi.org/10.1006/jmre.2001.2364>, 2001.
- Solyom, Z., Schwarten, M., Geist, L., Konrat, R., Willbold, D., and Brutscher, B.: BEST-TROSY experiments for time-efficient sequential resonance assignment of large disordered proteins, *J. Biomol. NMR*, 55, 311–321, <https://doi.org/10.1007/s10858-013-9715-0>, 2013.
- Sprangers, R. and Kay, L. E.: Quantitative dynamics and binding studies of the 20S proteasome by NMR, *Nature*, 445, 618–622, <https://doi.org/10.1038/nature05512>, 2007.
- Stallman, R. M. and GCC Developer Community: Using the GNU Compiler Collection, Free Software Foundation, Boston, 2012.
- Vallet, A., Favier, A., Brutscher, B., and Schanda, P.: ssNMRLib: a comprehensive library and tool box for acquisition of solid-state nuclear magnetic resonance experiments on Bruker spectrometers, *Magn. Reson.*, 1, 331–345, <https://doi.org/10.5194/mr-1-331-2020>, 2020.
- Veshtort, M. and Griffin, R. G.: High-Performance Selective Excitation Pulses for Solid- and Liquid-State NMR Spectroscopy, *Chem. Phys. Chem.*, 5, 834–850, <https://doi.org/10.1002/cphc.200400018>, 2004.
- Yang, D. and Kay, L. E.: TROSY Triple-Resonance Four-Dimensional NMR Spectroscopy of a 46 ns Tumbling Protein, *J. Am. Chem. Soc.*, 121, 2571–2575, <https://doi.org/10.1021/ja984056t>, 1999.
- Yong, J. R. J., Kupče, E., and Claridge, T. D. W.: Modular Pulse Program Generation for NMR Supersequences, *Anal. Chem.*, 94, 2271–2278, <https://doi.org/10.1021/acs.analchem.1c04964>, 2022.
- Zawadzka-Kazimierczuk, A.: New methods of protein NMR spectra analysis using the techniques of high dimensionality, Doctoral dissertation, University of Warsaw, 2012.